ANL/NE-15/42

Full Core Multi-Physics Simulation with Offline Core Deformation

Nuclear Engineering Division

ANL-NE-15/42

Full Core Multi-Physics Simulation with Offline Core Deformation

prepared by
E. Merzari, E. Shemon, Y. Yu, J. Thomas
Nuclear Engineering Division, Argonne National Laboratory

A. Obabko, R. Jain, V. Mahadevan,
Math and Computer Science Division, Argonne National Laboratory

J. Solberg, R. Ferencz, R. Whitesides
Methods Development Group, Lawrence Livermore National Laboratory

December 21, 2015

ABSTRACT

The NEAMS Reactor Product Line effort aims to develop an integrated multi-physics simulation capability for the design and analysis of future generations of nuclear power plants. The Reactor Product Line code suite's multi-resolution hierarchy is being designed to ultimately span the full range of length and time scales present in relevant reactor design and safety analyses, as well as scale from desktop to petaflop computing platforms.

In this report, building on previous reports issued in FY13 we describe our continued efforts to integrate thermal/hydraulics, neutronics, and structural mechanics modeling codes to perform coupled analysis of a representative fast sodium-cooled reactor core. The focus of the present report is a full core simulation with off-line mesh deformation.

Over the past five years, the Reactor Product Line effort has developed high-fidelity single-physics codes for neutron transport modeling, in the PROTEUS code, and computational fluid dynamics thermal/fluid modeling in the Nek5000 code. Both these codes have been exercised on over 100,000 processors of the IBM Blue Gene/P. The Diablo code has been used to perform structural mechanics and thermomechanical modeling. MOAB, the Reactor Geometry Generator (RGG), and MeshKit have been developed to generate and manipulate mesh and mesh-based data, in both serial and parallel environments. These tools together form a strong basis on which to build a multiphysics modeling capability. The goal of developing such a tool is to perform multiphysics neutronics, thermal/fluid, and structural mechanics modeling of the components inside a reactor core, the full reactor core or portions of it, and be able to achieve that with various level of fidelity. This flexibility allows users to select the appropriate level of fidelity for their computational resources and design constraints. We also note that while the focus of this report is on modeling a fast sodium-cooled reactor, another goal is that this simulation tool be useful for most reactor types.

Here we report on the continued integration effort of PROTEUS, Nek5000 and Diablo into the NEAMS framework. As compared with the FY13 report, the multiphysics setup has been updated to deal with mesh deformation due to the thermal expansion. Both codes have been demonstrated on a model of the Advanced Burner Test Reactor (ABTR). The reactor has been chosen, as opposed to the previously simulated EBRII reactor because of the recent features of the core design, including the core restraint system.

In fact, to design an inherently safe sodium-cooled fast reactor (SFR), it must be demonstrated that the net reactivity coefficient is negative, such any event that causes the core power to increase initially will be quickly followed by a response that tends to decrease the core power and return the reactor to a safe operating condition. This response in the core reactivity is caused by several mechanisms (which may compete with each other), including coolant density changes, the fuel Doppler effect, and changes in core geometry. Simulating the latter mechanism, changes in core geometry, is the focus of the multi-physics demonstration in this report. In particular, the focus is on the focus of radial core expansion caused by the motion of fuel assemblies in response to thermal expansion.

The core restraint system must be carefully designed to ensure that temperature increases induce thermal expansion that results in the fuel assemblies moving outward and thus away from each other. In particular, any fuel assembly distortion and displacement must occur in such a way that the fuel elements—typically located in the central portion of a significantly longer fuel assembly—move away from each other. This induces a negative reactivity response and helps return the reactor to a safe operating condition. In fact, in reactor designs under consideration by the Advanced Reactor Concepts program, core radial expansion is the dominant negative reactivity feedback mechanism.

This expansion phenomenon, which includes the physics of neutronics, thermal hydraulics, and structural mechanics, is challenging to model. In fact, conventional SFR safety analyses do not include the effect of fuel assembly bowing, and the physics are quite loosely coupled in a multi-step simulation procedure. The objective here is to demonstrate a multi-physics modeling and simulation capability that can explicitly predict the deformed core geometry, neutronics feedback with consistent power distributions, and temperature and flow distributions. This report describes the use of SHARP to perform a first-of-a-kind analysis of the core radial expansion phenomenon in an SFR for a full core.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 Introduction

SHARP [1], developed under the NEAMS program, is an advanced modeling and simulation toolkit for the analysis of nuclear reactors. SHARP is comprised of several components, including physical modeling tools, tools to integrate the physics codes for multi-physics analyses, and a set of tools to couple the codes within the MOAB [2] framework. Physics modules currently include the PROTEUS [3] neutronics code, the Nek5000 [4] thermal-hydraulics code, and the Diablo [5] structural mechanics code.

The development philosophy for the physics modules is to incorporate as much fundamental physics as possible, rather than developing tools for specific reactor analysis applications. This empowers designers to analyze transformative reactor concepts with simulation tools that are not limited to available experimental data sets from currently existing reactor designs. By developing the tools to be highly efficient on parallel computing platforms, employing millions of processor cores, engineering-scale simulations become practical on high-performance computers currently available at the DOE complex. Development efforts strive to work in tandem with efforts in experimentation, so that the tools are validated to produce accurate results for modeling physical phenomena that have been identified as important for nuclear reactor analysis. By taking this approach, SHARP supports nuclear reactor analysis and design activities for DOE programs and industrial partnerships with trustworthy modeling and simulation tools.

Previous work has demonstrated how the SHARP suite can be used to simulate single and multiple assemblies at different levels of resolution. Here we describe the continued progress in assembling of the SHARP Integrated Performance and Safety Code suite and its use in multiphysics modeling of a sodium fast reactor (SFR) full core in steady-state mode. In particular we focus in this report on the inclusion of mesh deformation due to the displacement produced by the thermal expansion.

The long-term goal is to simulate safety transients for the Sodium fast reactors (e.g., the Shutdown Heat Removal Tests of the EBR-II Experimental Breeder Reactor). The focus of the present simulation

The SHARP multi-physics modeling capability is being demonstrated for the problem of radial core expansion and bowing in a sodium-cooled fast reactor. To design an inherently safe fast reactor, reactivity dependence on radial core expansion must be engineered into the reactor plant to assure a loss of reactivity during transient events. In the advanced SFR concepts currently under consideration by the Advanced Reactor Concepts program, the core is designed to bow outward in response to thermal expansion of the structures in any transient where the core is heating. The grid plate and load pads, which support the core from below and restrain it from the top, respectively, also expand outward. Moreover, the core restraint system is designed such that the fuel assemblies bow outward in the middle, further separating the fuel pins. When controlled correctly, core expansion causes the fuel assemblies to move farther apart from each other, which has a negative reactivity effect and helps to shut down the reactor. Simulation of this expansion, which is essential to the safety of these reactor concepts, necessitates the coupling of structural mechanics, thermal hydraulics, and structural mechanics. This problem is very important to the

Advanced Reactor Concepts program, and it was selected as one of the driving problems for the Reactor Product Line.

The SHARP analysis that will be described in this report is a first-of-a-kind effort to perform a single integrated simulation of the core undergoing radial thermal expansion while retaining the full geometric detail necessary to model physical phenomena at the continuum scale. This simulation employs the neutronics, thermal-hydraulics, and structural mechanics simulations simultaneously, with no need for offline perturbation analyses or passing coefficients between the physics codes. The reactivity feedback from core deformation is being predicted explicitly. The power distribution from nuclear fission is predicted by the neutronics module, which influences the temperature field in the thermal-hydraulics module. Structure temperature profiles predicted by the thermal-hydraulics module drive deformations in the structural mechanics module, and the neutronics and thermal-hydraulics simulations are repeated on the deformed core geometry provided by the structural mechanics simulation. SHARP is uniquely posed to perform this simulation, as PROTEUS is the only deterministic neutronics code with an unstructured grid capable of solving the neutron transport equation on a highly complex deformed geometry. The complexities of mesh mapping and motion, solution transfer, and parallel efficiency are well suited for SHARP's sophisticated integration tools.

Therefore, the effort reported here has two objectives: (1) to develop and demonstrate the capability to model core radial expansion phenomena with SHARP and (2) to evaluate the feasibility of modeling this phenomena for large-scale full-core applications. Such large-scale structural deformations have only recently been performed with SHARP [6], and some code development efforts were required to deform the geometry, smoothen the discretized mesh, and communicate these changes in a fashion suitable for the three physics modules.

A model of the Advanced Burner Test Reactor (ABTR), with 199 assemblies, was developed for each of the three physics modules. After surveying several options, the ABTR was chosen as the target demonstration problem for the following reasons: the design of ABTR incorporates structural mechanical feedback by assembly bowing, and information on the ABTR is readily available, unlike other facilities like the Fast Flux Test Facility.

The remainder of this report is organized as follows:

- In *Section 2* the ABTR core and the problem of radial core expansion is described in detail.
- In *Section 3* we give background on the single-physics codes used and the infrastructure used to integrate them into a multiphysics capability. In particular this section described additional development performed in Nek5000 to handle this effort. In particular a new porous medium model has been developed and verified to produce excellent results in predicting the temperature in the assembly ducts.
- In *Section 4* the results of the multiphysics calculation are discussed in detail.

2 Problem Specifications

Detailed specification of the ABTR problem is provided in a separate companion report [5], and therefore is only briefly summarized here.

The ABTR is a conceptual advanced sodium-cooled fast reactor (SFR) designed by Argonne to serve as a prototype capable of demonstrating the technology of burning high-actinide fuel while simultaneously producing electricity [31]. ABTR is rated for a thermal power of 250 MW with an electric output of approximately 95 MW. As in most SFR designs, fuel subassemblies consist of an array of pins supported within a thin-wall hexagonal duct (Figure 2.1). The assemblies, which have a hexagonal cross section, are inserted in a lattice (Figure 2.2). The 199 assemblies can be categorized and counted as 54 driver fuel assemblies, 78 reflector assemblies, 48 shield assemblies, 10 control rod assemblies, 6 fuel test assemblies, and 3 material test assemblies. The 54 driver fuel assemblies are categorized into 24 inner zone driver assemblies and 30 outer zone driver assemblies. The inner zone driver assemblies have lower TRU enrichment (16.5%) than the outer zone assembly (20.7%), which helps to maintain a flattened power distribution. All ABTR assemblies have the same HT-9 hexagonal duct structure, SS-316 lower structure, and upper handling socket. Sodium flows through the gaps between assemblies.

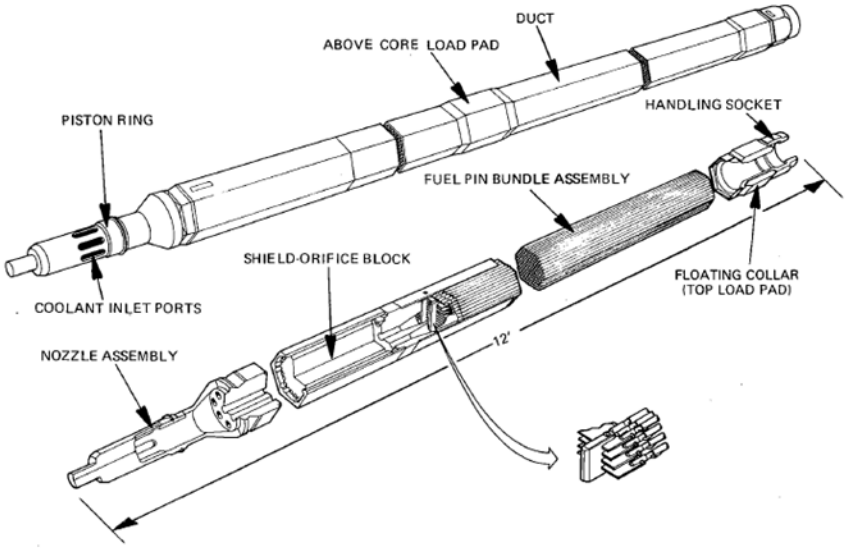


Figure 2.1 Typical SFR fuel assembly.

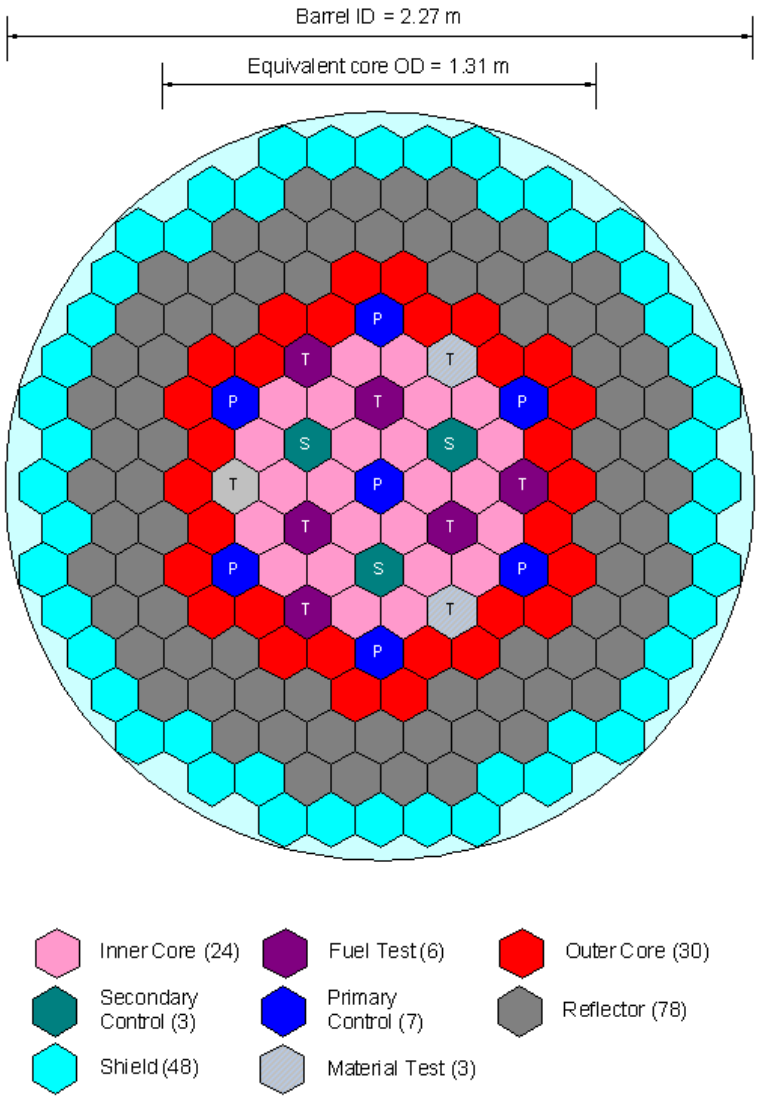


Figure 2.2. ABTR full-core assembly layout.

To ensure the negative reactivity response associated with bowing of the fuel assemblies, the ABTR utilizes the "limited free bow" core restraint system (Figure 2.3). The restraint system is characterized by top load pads (TLPs) on the assembly ducts at the top and above-core load pads (ACLPs) in the region above the core, along with restraining rings at the TLP and ACLP axial heights. The rigid restraint rings are attached to the core barrel at the ACLP and TLP locations. The load pads serve as preferential contact points between the ducts. The pads add only marginal thickness to the main duct body (thickness is exaggerated for clarity in the figure) but they are nonetheless thick enough to maintain the desired form under the design loadings. Additionally, the design ensures that duct-to-duct loading (resulting from bowed ducts in contact) is kept within allowable limits, including the time-dependent inelastic bowing effects due to irradiation (and thermal) creep and swelling effects.

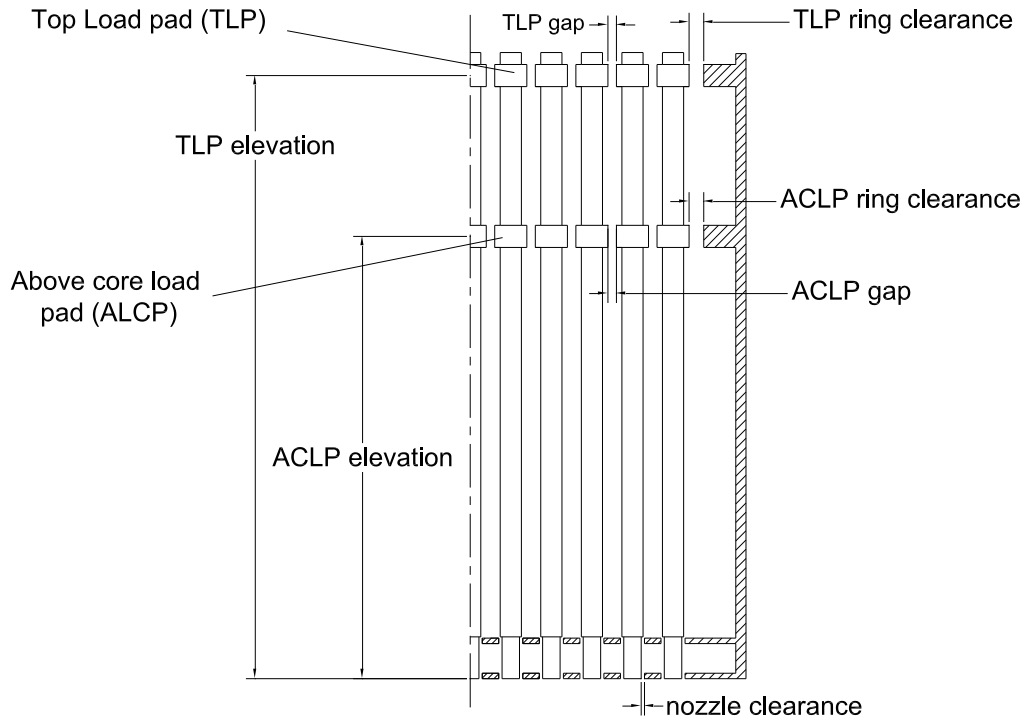


Figure 2.3. Limited free bow core restraint system.

The limited free bow core restraint system is designed to provide inherent protection against over power events by taking advantage of thermally induced bending action of the fuel ducts. This is illustrated in Figure 2.4, which shows a row of three cantilevered ducts located symmetrically about the center of a core and in a radially varying thermal gradient. Figure 2.4a shows the nominal configuration of the ducts with no temperature gradient. As the radial thermal gradient develops (temperature increases as distance from centerline decreases), the ducts begin to bow outward as shown in Figure 2.4b. Prior to contact with the top core restraint ring, the duct bends away from the core centerline as the temperature increases and therefore reduces the reactivity insertion. After contacting the top restraint ring and as the temperature gradient increases, the center of the duct bows inward which temporarily increases the reactivity. As the gradient increases, the inward bowing continues until the ducts contact at the ACLP. When the interior ducts all contact at the ACLP, the reactor is 'locked-up' and no further compaction can occur. Subsequent increased thermal gradients cause a reverse bowing below the ACLP moving the core region away from the core center as illustrated in Figure 2.4c. At this point, the reactivity generally decreases with constant negative slope as temperature increases. The core restraint system is designed to have this lock-up occur below the nominal operating core outlet temperature. In this way, the core is already locked up during normal operation, and any transient events with increasing temperature will induce further outward bowing in the middle of the core.

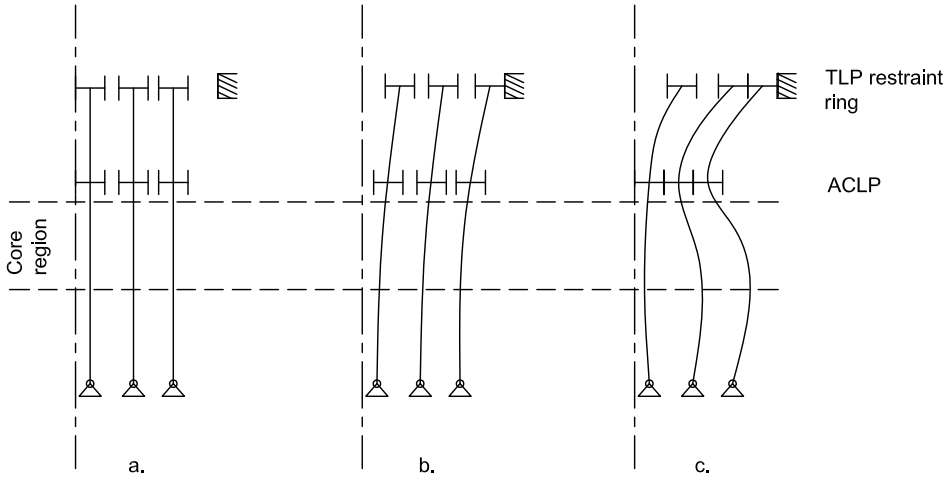


Figure 2.4. Operating mechanism of the limited free bow core restraint.

3 Overview of SHARP and Computational Models

The NEAMS Reactor Product Line (RPL) aims to develop an integrated multi-physics simulation with a multi-resolution hierarchy that is designed to ultimately span the full range of length and time scales present in relevant reactor design and safety analyses, as well as scale from desktop to petaflop computing platforms. This section discusses the design and the numerical methodologies used in the SHARP toolkit to integrate neutronics, thermal-hydraulics, and structural mechanics physics components to perform coupled reactor analysis on a representative SFR core geometry. Based on the requirements specified, a problem to quantify the primary structural mechanical feedback effect with multi-way coupling has been implemented with dual resolution: a detailed heterogeneous model represents the duct surrounding each assembly while interior of the ducts (the individual assemblies) are represented with a homogenized geometry.

In order to produce a fully coupled-physics simulation capability, two obvious approaches can be pursued. In one approach, existing single-physics codes/components can be assembled into an overall coupled simulation code with appropriate interfaces to communicate between the components to capture the nonlinear feedback effects. This is generally referred to as a "small-f" or "bottom-up" framework approach [1, 8]. The other approach is to use an integrated, coupled-physics modeling framework, with new code pieces for each relevant physics area developed inside that framework from scratch. This is sometimes referred to as a "large-F" or "top-down" approach [9, 10]. The primary advantage of the former approach is that it preserves several man-years invested in existing verified and validated individual physics modeling codes, but at the cost of some intrusive modifications to enable the software interfaces. The large-F approach avoids intrusive interfacing by providing a unified platform to enable coupling, but at the cost of re-writing all the necessary physics codes and verifying the components individually and as a whole. The overall approach being pursued in the RPL effort is to develop and demonstrate a small-f framework for performing coupled multiphysics analysis of reactor core systems. This system takes advantage of many single-physics codes also sponsored by the overall NEAMS program over past several years.

This relevant detail regarding the background on construction of the RPL coupled-physics framework (SHARP) along with the methodology is discussed in the following sections.

3.1 The SHARP Multi-physics Code System

A multi-physics reactor core modeling code can be constructed in many ways, and numerous past efforts have provided stepping-stones for future efforts [10]. What distinguishes the SHARP effort from others is the goal of flexibility in the physics, discretization types, and software options supported by the framework. This section describes the SHARP modeling approach in detail and illustrates how various existing physics codes have been connected to this framework.

As stated above, SHARP employs a "bottom-up" approach, so it can use existing physics codes and take advantage of existing infrastructure capabilities in the MOAB framework and the coupling driver/solver library, the Coupled Physics Environment (CouPE), which utilizes the widely used, scalable PETSc library [11].

Using an existing physics code in this system requires that the system support the mesh type used by the individual physics models. The physics models can retain their own native representation of the mesh, which gets transferred to and from MOAB's representation through a mesh adaptor; or it can use MOAB's representation directly. Language interoperability through the C/Fortran-based iMesh interfaces also allows flexibility in the implementations that are tuned to individual physics requirements without overhead.

In practice, this means that the coupled system may be solved on multiple meshes, each of which models part or all of the physical domain of the problem. To perform efficient coupled calculations, the results must be transferred from the mesh on which they are generated (source mesh), to the mesh for which they provide initial or boundary conditions (target mesh) due to nonlinearity introduced because of coupling between physics models. "Multi-way" transfer is required in cases where the physics depend on each other's solution fields, for example in reactor analysis where neutronics computes heat generation based on temperature properties computed by thermal-hydraulics, which in turn depends on the heat-generation source term computed by neutronics.

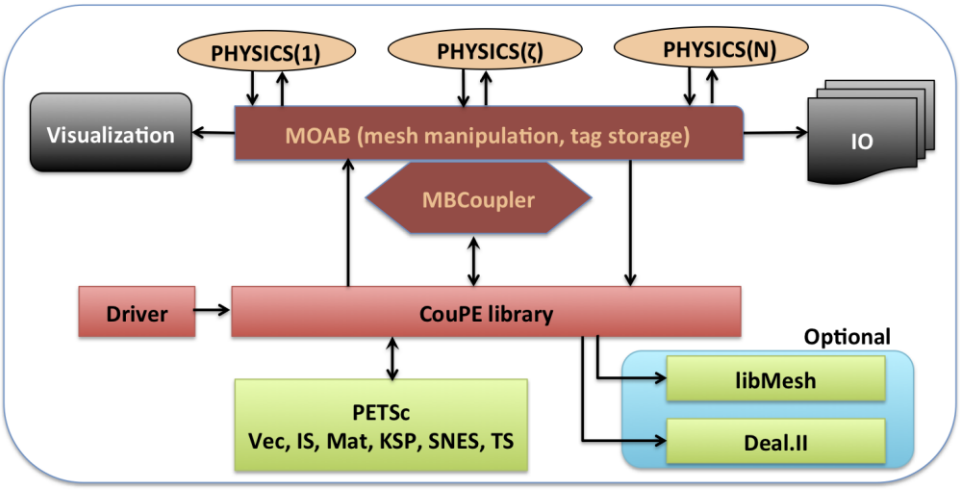


Figure 3.1. SHARP architecture.

Since relevant physics components solving a nuclear engineering problem have widely varying backgrounds in terms of code architectures, dependency requirements, and specialized solver data-structures, a flexible approach to the coupling methodology was necessary to obtain accurate solutions. This motivation led to the development of the MOAB-based spatial projection tools and the CouPE drivers based on PETSc library to orchestrate the global nonlinear solver. Details regarding these tools are given in the following sections.

3.1.1 Mesh Database (MOAB)

One of the critical aspects in assembling a multi-physics modeling code is mapping the results from one physics domain to another. In the small-f RPL framework, a common mesh library serves this purpose. The MOAB library provides a "data backplane" to link physics models through their spatial domains, and MOAB's MBCoupler package transfers physics results between those domains.

MOAB is a generic library for query and modification of structured and unstructured mesh and field data associated with the mesh [2]. MOAB can represent all entities typically found in the finite-element zoo, as well as polygons, polyhedra, and structured meshes. MOAB provides parallel functionality for resolving entity sharing and ghosting between processors, with sharing and ghosting information available as annotations on the local mesh. MOAB's parallel I/O is based on the parallel HDF5 library, and it has been demonstrated on processor counts up to 16,000 (on the IBM BlueGene/P system). A partitioning tool has been implemented by interfacing with the Zoltan partitioning library implementation, with in-situ visualization provided by a Paraview plugin. MOAB can read meshes generated by the CUBIT mesh generation toolkit and can represent the various mesh types used in this effort.

MOAB's data model consists of four fundamental data types:

- Entity: A basic entity in the discrete model, e.g., vertex, quadrilateral, tetrahedron.
- Entity Set: An arbitrary collection of entities.
- Tag: A piece of data annotated on entities and sets.
- Interface: The primary database object instance.

Even though this data model seems simple, it can represent all the necessary data to run coupled simulations. In particular, tags can be used to store both fine-grained solution data on individual vertices and elements and coarse-grained annotation of sets to identify them as boundary conditions, material types, or processor partitions. Two particular groupings are common to this effort:

- Material sets, also referred to as "element blocks," group elements by material definition. In MOAB, these are represented as entity sets, tagged with a "MATERIAL SET" tag whose value stores a user-assigned id number.
- Neumann sets, also referred to as "sidesets," store groups of lower-dimensional entities (in a 3D mesh, Neumann sets contain mesh faces, for example). Similarly, these groups are marked with "NEUMANN SET" tag whose value is the user-assigned id number.

Other tags used to store field and other data for PROTEUS and Nek5000 are described in sections 3.2 and 3.3, respectively.

3.1.2 Solution Transfer Tool (MBCoupler)

MBCoupler, a MOAB-based tool for solution transfer [12], has been demonstrated on up to 4,000 processors. This tool allows the source and target meshes to be distributed across processors in whichever way are best suited for the physics associated with each mesh. Target-to-source mesh point location is performed in parallel, with bounding-box-based acceleration used to determine possible source mesh processors containing every point and with KD-Tree decomposition used locally on each processor. This tool can transfer solutions using both linear finite element and piecewise-constant shape functions. As described in Section 3.4, it has recently been extended to incorporate spectral element shape functions as well. In the demonstration described in this

report, MBCoupler is used to map the results computed by one physics module to boundary conditions on the mesh used by the next physics module.

3.1.3 Coupled Physics Environment (CouPE)

CouPE provides scalable and extensible interfaces to couple different physics components that are nonlinearly dependent on each other. The SHARP multiphysics coupled code for reactor analysis problems employs validated and verified efficient single-physics codes with message passing interface architecture to achieve tight coupling with an iterative operator split methodology [10, 13]. Such iterative nonlinear methods provide the flexibility to use standard industrial codes and avoid replicating man-years of development and testing by following the bottom-up approach (section 3.1).

The aims in designing the CouPE code library included the following:

1. Make use of existing libraries and physics codes to minimize development time and base the framework on already well-verified and validated single-physics codes and libraries.
2. Enable a flexible and accurate data exchange framework between codes in a mesh, numerics, and physics aware fashion, i.e., maintain consistency, accuracy, and conservation of key fields.
3. Provide flexible data containers and physics objects to facilitate and simplify the evaluation of the non-linear residuals representing the fully discrete partial differential equations for different physics components.
4. Employ different kinds of multi-physics coupling strategies within the same architecture with minimal changes in the driver.
5. Enable runtime object polymorphism.

CouPE aims to solve all of the physics components under a unified framework in order to exchange the solution from one physics model to another and converge the coupled-physics solution fields to user-specified tolerances without sacrificing numerical stability or accuracy. CouPE provides the necessary components and layers to wrap existing physics codes or write a complete description of a physics problem from scratch to solve phenomena of interest, that is to enable both bottom-up and top-down approaches. The library also provides the necessary tools to quickly implement any of the popular variations of an operator-split coupled solver (Marchuk, Strang, Yanenko among others [13]) or a more rigorous matrix-free inexact-Newton solver with a Jacobian-free Newton-Krylov (JFNK) technique [14]. Currently, the Marchuk splitting with Picard iteration over the physics components has been implemented in CouPE; other coupling strategies have been implemented but need to be tested for relevant multi-physics problems.

For stationary, coupled nonlinear problems, the primary source of error stems from the exchange of physics solutions that reside in different spatial discretizations and resolutions. CouPE utilizes the iMesh interfaces. And, more specifically, its implementation by MOAB and MBCoupler enable seamless integration of the single-physics codes. This is made possible by exposing a minimal interface to be implemented by the physics wrappers, whose design follows the software paradigms of PETSc [15]. The current design of CouPE is intended to satisfy the need for a

loosely coupled software framework to solve strongly coupled physics modules. The implementation of coupled methods is usually difficult and CouPE can reduce the development time by providing a template to solve a collection of nonlinearly coupled physics objects via a uniform interface. The driver is simple, transparent, and extensible. It can be thought of as a "glass-box" solver rather than a "black-box" solver, since it provides access to all the internal details of the physics and the corresponding internal mesh structures, and it allows the user to supply and override the behavior at runtime.

Similar to the PETSc toolkit library, CouPE is designed to allow the user to specify command-line arguments to control the dynamic behavior of the coupled solver. The parameter specifications include the input for individual physics components, input mesh parameters, and type of the solver, and in advanced usage, can even dynamically change the type of the physics being coupled. This is made possible by completely abstracting the behavior of the core object until runtime, even though the internals of these objects are fully available to the driver. Hence, the core implementation of a physics object is hidden while the driver utilizes only the methods exposed in the public interface. The advantage of such a method is that the implementation of the coupled physics driver and the accompanying physics components need to be compiled, linked, and verified only once and then can be reused in a variety of different coupling methods (e.g., loose versus tight coupling).

3.2 SHARP Physics Components

In the SHARP framework, MOAB interfaces are implemented for 3 different physics components that are relevant to fast reactor physics analysis. The addition of a new physics component to the framework requires integration and ability to read the mesh and possibly associated data from iMesh/MOAB formats, along with implementation to propagate solution variables back onto the mesh after their computation via tags defined either on discrete vertices or elements. Because of the various storage formats used in physics models, and the parallel domain-decomposed environment in which these calculations are usually run, this integration process can be somewhat involved.

To better understand the level of fidelity that can be achieved by the SHARP framework, some key aspects of the 3 physics components are given below.

3.2.1 Neutron Transport Solver (PROTEUS)

PROTEUS is a high-fidelity deterministic neutron transport code based on the second-order even-parity formulation [16]. The application scope targeted for PROTEUS ranges from the homogenized assembly approaches prevalent in current reactor analysis methodologies to explicit geometry approaches, with the ability to perform coupled calculations to thermal-hydraulics and structural mechanics. The PROTEUS solver has a proven capability of using existing petascale parallel machines to solve problems with demonstrated scalability of over 70% (strong scaling) at over 250,000 processors (on BlueGene/P). These achievements of PROTEUS were made possible by partitioning the space-angle system of equations over the available processors and utilizing established iterative solution techniques from the neutron transport community combined with the parallel algorithms in the PETSc toolbox.

Interfaces to the MOAB mesh database have been written to handle UNIC meshes that describe detailed geometries with multiple blocks (regions) with appropriate specification hooks for temperature- and density-dependent material cross-section evaluation and interpolation. This interface is essential to capture the nonlinear feedback effect from thermal-hydraulics. Inherent ability to use a deformed mesh with appropriate recalculation of the density changes within materials (thereby affecting cross-sections) have also been implemented to enable direct coupling to a deformation code such as Diablo.

The eigenvalue solver in PROTEUS computes the neutron flux shape, computes the power distribution in the reactor, and then places the computed data in appropriate MOAB tags. The power solution field is then propagated to the other physics solvers via the data-coupling interfaces that support tight coupling with thermal-hydraulics, which uses the tag data as a thermal source term to compute temperature fields. Several verification studies have been performed during the quality assurance process to ensure that the coupled solver solutions are physically meaningful.

3.2.2 Computational Fluid and Thermal Dynamics Solver (Nek5000)

The Nek5000 computational fluid dynamics solvers are based on the spectral element method developed by Patera [17]. Nek5000 supports two different formulations for spatial and temporal discretization of the Navier-Stokes equations. The first is the P_N-P_{N-2} method with velocity/pressure spaces based on tensor-product polynomials of degree N and $N-2$ respectively. The second is the low-Mach number formulation of Tomboulides and Orszag [18], which uses consistent order-N approximation spaces for both the velocity and pressure. The low-Mach number formulation is also valid at the zero-Mach (incompressible) limit [19]. The Nek5000 code has been extensively verified and validated for several benchmark problems and has a proven scalability in existing petascale architectures up to 131,072 processors (over a billion degrees-of-freedom).

The conjugate heat transfer problems that are typically present in nuclear engineering applications can be solved rigorously using the formulations in Nek5000. Typically, the following boundary conditions are applied at the inflow, outflow, and wall surfaces:

- The inlet surface has uniform prescribed velocity and fixed temperature,
- The outlet surface has standard outflow boundary conditions, and
- The wall surfaces have velocity non-slip boundary conditions.

Using the standard iMesh-based interfaces to MOAB, several different mesh formats can be natively used with Nek5000 along with the extended ability to couple with other physics components in the SHARP framework. When running Nek5000 in the fully coupled mode, the fluid/solid temperatures, along with their corresponding densities, are stored in MOAB tags to be used either by the structural mechanics or neutronics components, propagated by the unified data-transfer mechanisms detailed earlier. The MOAB interface also enables Nek5000 to perform application checkpointing and have restart capabilities independent of the number of processors, thereby enabling the user an opportunity to investigate the validity of the coupled solution before proceeding further.

In order to perform thermofluid analyses of homogenized fuel assemblies, a porous media model was implemented in the Nek5000 code. The porous media model, sometimes referred to as a distributed resistance model, is based on the model implemented into the STAR-CD code [16]. Porous media models are typically applied to problems where the fluid flows through a region with many small-scale solid structures, and it would be impractical to resolve the geometry explicitly. Instead, the effect of the small-scale solid structures on the flow is modeled as a momentum sink or resistance in a homogenized fluid domain. In this particular case, we wish to model the influence of the fuel pins on the flow, i.e., drag and pressure drop, as a momentum sink without explicitly representing the geometry of thousands of fuel pins. The model must also account for the energy deposition associated with nuclear fission. Moreover, fuel and cladding temperatures are estimated for each fuel assembly, and may be provided to the neutronics code in future coupled simulations.

Because the porous media model employs a fairly standard model, which may be found in the STAR-CD manuals [20] among other sources, it is only summarized here. First, a volume porosity χ is defined as the ratio of open volume to total volume of the porous medium. This is used in all time-derivative terms in the mass, momentum, and energy continuity equations to provide the appropriate fluid inertia.

An additional body force per unit volume is added to the momentum equation such that:

$$\boldsymbol{F}_p = -K \cdot \mathbf{v} \tag{1}$$

where K is the porous resistance tensor and $\mathbf{v}$ is the superficial velocity. Superficial velocity is defined as the volumetric flow rate divided by the total cross-sectional area. For each of the three directions (i = 1,2,3), K is a diagonal matrix given by:

$$K_{ii} = \alpha_i[\mathbf{v}] + \beta_i \tag{2}$$

where α_i and β_i are model-dependent coefficients with dimensions of [mass × length^{-4}] and [mass × length^{-3} × time^{-1}], respectively. Note that the repeated indices denote the diagonal elements of the tensor, not summation. For channel flow, the resistance may be considered orthotropic, i.e., only causing resistance in the Z-direction. An appropriate choice of α and β can be determined from an empirical formulation:

$$\alpha = \frac{PC_1\rho Re^n}{2\chi^3 A}$$
$$\beta = \frac{PC_1C_2\mu Re^n}{2\chi^2 AD_h} \tag{3}$$

where C_1, C_2, and n are model coefficients; P is the wetted perimeter; A is the superficial area; D_h is the hydraulic diameter; ρ is the density; and μ is the molecular viscosity.

The influence of turbulence on the momentum transport is assumed to be included in the porous resistance term in Eq. (1). However, turbulence is included in the diffusional flux term in the energy transport equation in the fluid, which is conventionally:

$$F_{h_j} = -\lambda \frac{\partial T}{\partial x_j} + \bar{\rho}\overline{u_j' h'} \tag{4}$$

where λ is the thermal conductivity, T is the temperature, u_j' is the fluctuating component of velocity in the x_j-direction, and h' is the fluctuating component of enthalpy. The second term in this expression represents turbulent diffusion of thermal energy. This term is evaluated using the turbulent kinetic energy k and dissipation ϵ from the k-ϵ turbulence model:

$$k = \frac{3}{2} I^2 [\mathbf{v}]^2$$
$$\epsilon = \frac{C_\mu^{\frac{3}{4}} k^{\frac{3}{2}}}{L} \tag{5}$$

where I is the turbulence intensity, L is the turbulence length scale, and C_μ is a coefficient that equals 0.09 in the standard k-ϵ model. Each of these is strongly dependent on the model. These values may then be used to evaluate the turbulence viscosity ν_t using the following relation:

$$\nu_t = \frac{C_\mu k^2}{\epsilon} \tag{6}$$

Then given a value of the turbulent Prandtl number $\mathrm{Pr_t}$, the thermal diffusivity term can be evaluated.

Heat is generated by nuclear fission in the fuel, conducted through the fuel and cladding, and removed by the coolant. The fuel and cladding are both part of the solid portion of the domain, which is causing the flow resistance. An interphase heat transfer term is added to the coolant energy transport equation, which requires the cladding outer surface temperature $T_{clad,o}$.

$$Q_v = h_x \left(T_f - T_{coolant}\right) \tag{7}$$

where h_x is the thermal conductance per unit volume and $T_{coolant}$ is the coolant temperature of the permeating fluid. Energy transport equations for the solid components must also be written, with volumetric heat generation in the fuel and energy transfer from the fuel to the coolant that

matches Eq. (7). The equations are solved in a steady state and coupled fashion. In the following we describe efforts to improve the porous medium model.

3.2.2.1 A new porous media model

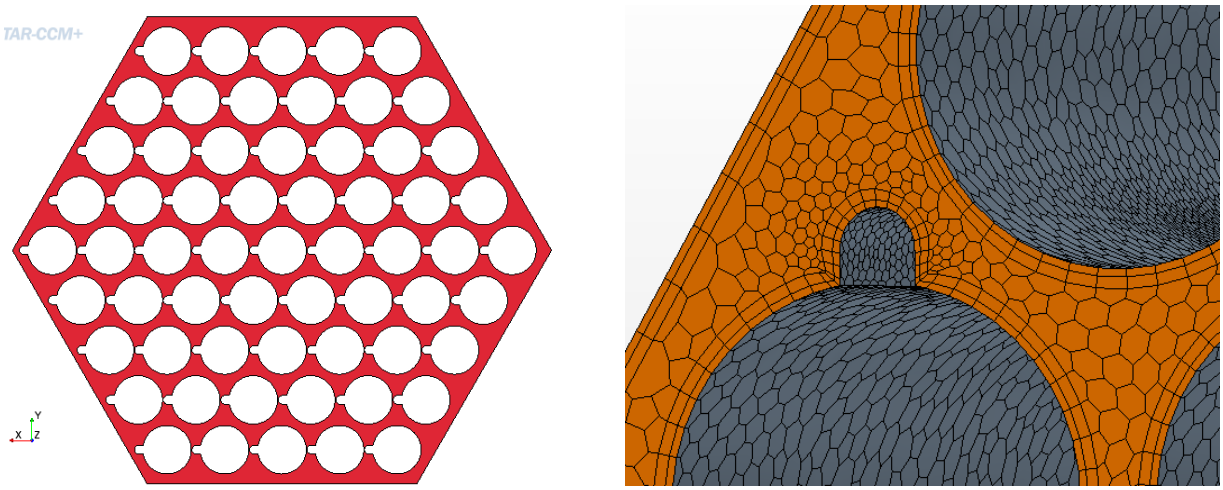
Porous medium modeling for wire wrapped rod bundles has long being applied [21], but it has no history of being applied to temperature predictions inside the assembly ducts. To evaluate porous media models for ducted fuel assemblies with wire-wrappers, reference simulations were performed with a CFD model. The geometry was explicit represented in the CFD model, with a 61-pin wire-wrapped fuel bundle. The software used for the comparison is the commercial CFD code STAR-CCM+. The reference simulation was compared to various models that employ porous media. This work proposes three porous medium models (i.e., uniform model, 2 region model and 3 region model) based on a theoretical analysis. The results obtained from 3 region model were found to predict the duct temperature with as good a precision as do the results from CFD simulation. This preliminary work only deals with assemblies, which operate under uniform power distribution. The effect of the power distribution in radial and axial direction will be investigated in future work.

Table 3.1. Detailed information on the 61 pin EBR-II 425 fuel assembly.

Parameter	*Value*	*unit*	*Parameter*	*Value*	*unit*
number of rods	61	-	assemble length	36.652	cm
rod diameter	5.84	mm	inlet velocity	4.865	m/s
rod pitch	6.8912	mm	X span(±)	0.032389	m
P/D	1.18	-	Y span(±)	0.02805	m
wire wrap diameter	1.07	mm	hydraulic diameter	2.75E-03	m
wire wrap pitch	15.24	cm	Reynolds number	43552	-

The 61 pins EBR-Ⅱ425 fuel assembly is used as a reference fuel assembly design in this study. The assembly has a relative large coolant flow area with P/D of 1.18 and a relative short core length (only 2.25 spans of the wire wrap). As a result of the high inlet velocity (4.865 m/s), the reference case has a high Reynolds number (43552). In addition, the inlet flow temperature and the rod heat flux for the reference case is 664K and 1.74E6 W/m^2 respectively. More detailed information on the assembly is listed in Table 3.1. It is confirmed in previous studies [22-24] that RANS-based simulation can provide acceptable accurate hydrodynamic prediction in the wire wrap rod bundle assembly. The simplified modeling of the wire–pin contact and a polyhedral mesh of 20 million computation cells applied in the simulation are shown in Figure 3.2. A Realizable k-ε turbulence model with the two-layer all-y+ wall formulation is used in the

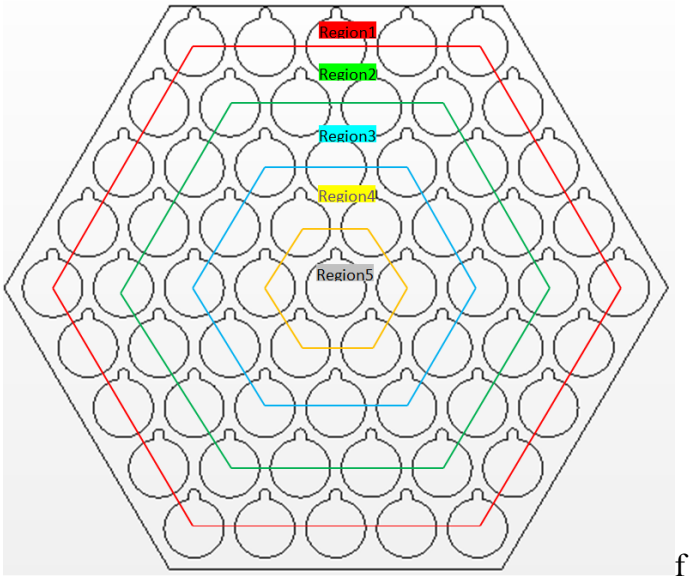
simulations. A segregated flow solver with the SIMPLE predictor-corrector algorithm is used in all calculations. The solution is well converged as the normalized residual are below 10^{-4}.



(a) Cross section of the wire wrap rod bundle

(b) Polyhedral mesh of the subassembly

Figure 3.2. Geometry and generated mesh of the 61 pin wire-wrap rod bundle



f

Figure 3.3. Sketch of five regions.

The assembly can be divided into 5 regions. The shape and the position of these five regions are shown in Figure 3.3. Region 1 is located at the outermost layer of the duct and Region 5 is located at the center of the duct. The rest regions are located in between these two regions with same space interval. Assuming uniform heat for each pin the average volumetric heat flux in each region is calculated with the following formula:

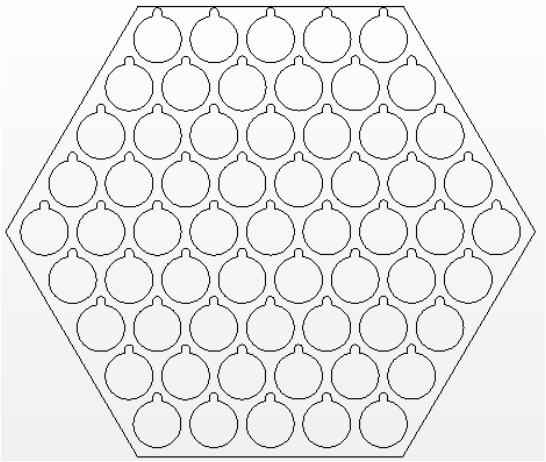
$$Q_v = \frac{N \cdot S \cdot Q_s}{V} \qquad (1)$$

where N is the rod number in each region , S is side area of each rod (m^2), Q_s is surface heat flux of each rod W/m^2,and V is the region volume. The detailed calculation parameters are listed in Table 3.2

Table 3.2. Parameters in five regions.

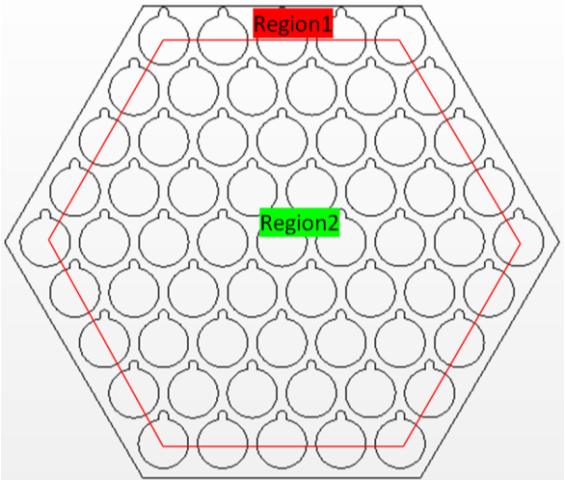
Parameter	*Region1*	*Region2*	*Region3*	*Region4*	*Region5*
Pins number	13	21	15	9	3
Surface Heat flux	Q_s	Q_s	Q_s	Q_s	Q_s
X span(±)	4p~X	3p~4p	2p~3p	1p~2p	0~1p
side area(S)	S	S	S	S	S
Height	H	H	H	H	H
Volume(V)	$\frac{\sqrt{3}}{2}(X^2 - 16p^2)H$	$\frac{21\sqrt{3}}{2}p^2H$	$\frac{15\sqrt{3}}{2}p^2H$	$\frac{9\sqrt{3}}{2}p^2H$	$\frac{3\sqrt{3}}{2}p^2H$
Volumetric heat flux (Qv)	$\frac{26\sqrt{3}}{3}\frac{Q_sS}{(X^2-16p^2)H}$	$\frac{3\sqrt{3}}{3}\frac{Q_sS}{p^2H}$	$\frac{3\sqrt{3}}{3}\frac{Q_sS}{p^2H}$	$\frac{3\sqrt{3}}{3}\frac{Q_sS}{p^2H}$	$\frac{3\sqrt{3}}{3}\frac{Q_sS}{p^2H}$

It is quite remarkable that the volumetric heat flux is region 1 is different from that in other regions. That's because the porosity of the porous medium which represents the rod concentration in the outmost layer is smaller than that in the center of the duct. Therefore, three models (i.e., uniform model, 2 region model and 3 region model) are proposed to investigate the effect of the volumetric power distribution based on the calculation above. In the uniform model, the whole assembly is taken as one porous medium region with uniform volumetric heat flux. In the 2 region model, the outmost layer in Figure 3.3 is taken as region 1 and the remaining regions are combined together and taken as region 2 due to their same volumetric heat flux. In the 3 region model, the outmost region in 2 region model is further divided into two regions which include a thin outmost region without implanting volumetric heat flux. The sketch of the regions in three models and volumetric heat source in each region are presented in Figure 3.4.



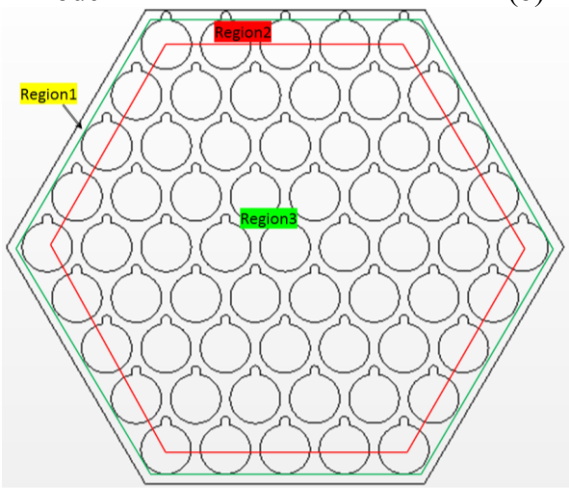
Q_v=7.75E+08

(a) uniform model



Q_{v_1}=5.99E+08(W/m^3) , Q_{v_2}=8.42E+08 (W/m^3)

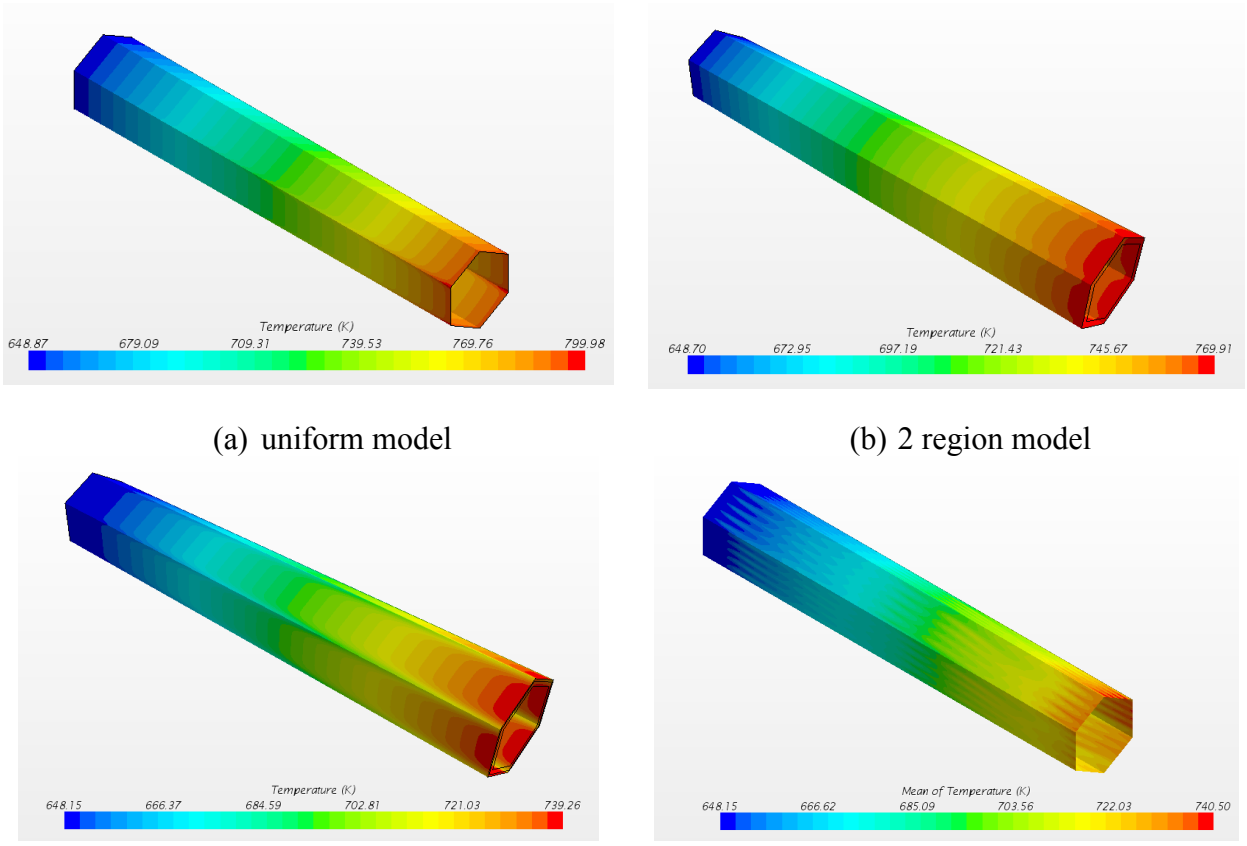
(b) 2 region model



Q_{v_1}=0 (W/m^3) , Q_{v_2}=8.79E+08 (W/m^3) Q_{v_3}=8.42E+08 (W/m^3)

(c) 3 region model

Figure 3.4. Sketch of the regions in three models and volumetric heat source in each region.



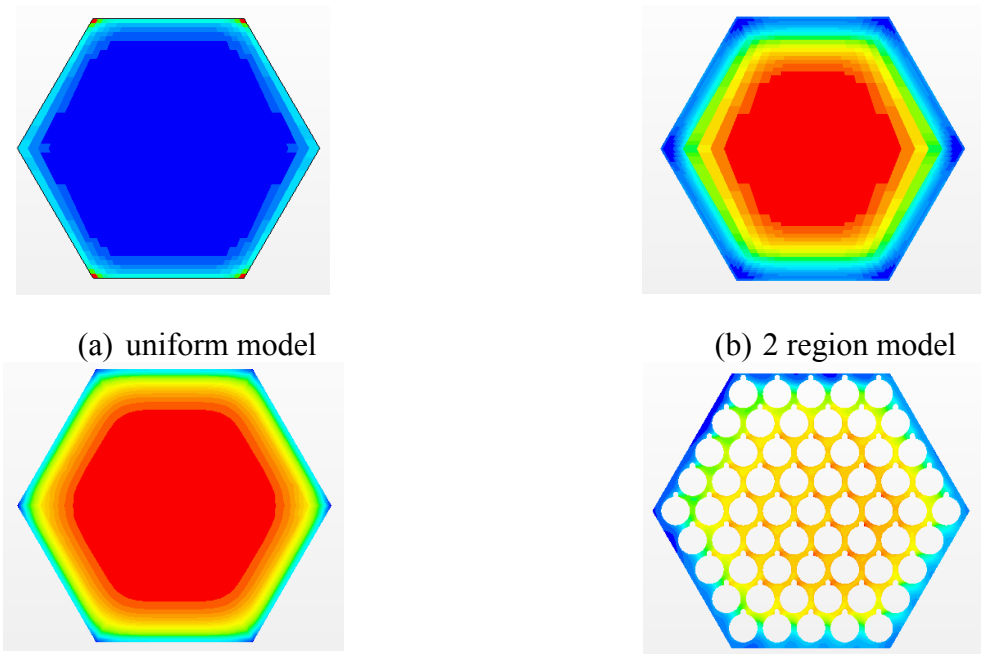
(a) uniform model (b) 2 region model

(c) 3 region model (d) CFD

Figure 3.5. Contour of the duct temperature distributions.

Figure 3.5 presents the duct temperature distribution obtained from different models. The duct temperature obtained from CFD calculation ranges from 648.2K to 740.5K. The duct temperature range obtained from porous medium model i.e., uniform model, 2 region model and 3 region model is 648.9K to 780K, 648.7 to 770K and 648.2K to 739.3K respectively. It is found that the uniform model and the 2 region model overestimate the duct temperature. The duct temperature range obtained from 3 region model agrees well that from the CFD model. The contour isoclines of the duct temperature in the CFD model present obvious serration which is due to different hydrodynamics diameters of the sub-channels close to the wall. Without taking into account that factor, the contour isoclines of the duct temperature from the porous medium model is relatively flat. However, the contour isoclines duct temperature on each wall from the uniform model and the 3 region model have different bend direction. The temperature distribution on the outlet shown in Figure 3.6 better explains this distribution. In the uniform model, the hot spot appears at the corner of the hexagonal duct. On the contrary, the cold spot appears at the same position in the 3 region model.

In order to analyze the duct temperature from different model quantitatively, the duct temperature from one side of the hexagonal assembly is extracted on three elevations: low (z=0.01m), center (z=0.2m), high (z=0.36m). The sketch of the locations is shown in Figure 3.7.



(a) uniform model (b) 2 region model

(c) 3 region model (d) CFD

Figure 3.6 Temperature Distribution on the outlet.

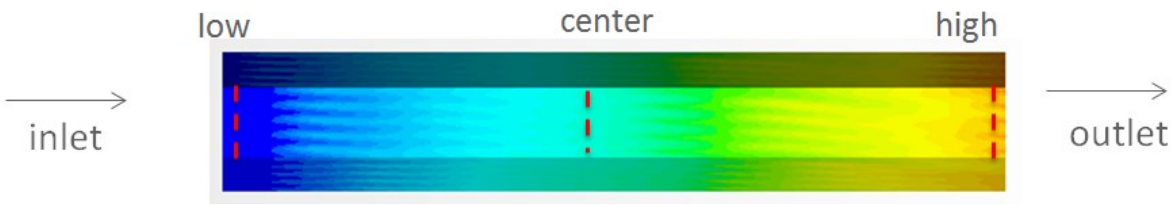


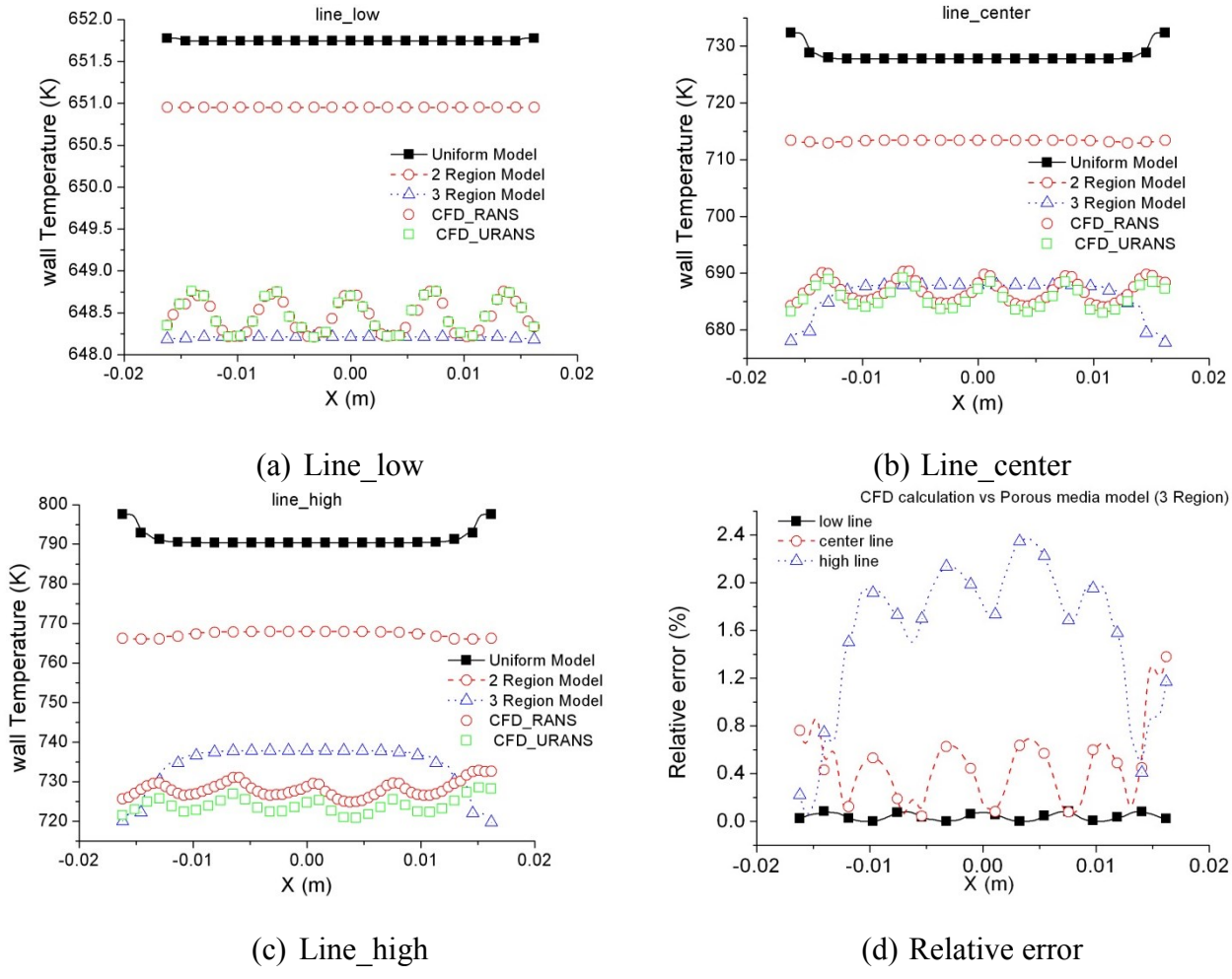
Figure 3.7. Sketch of the computational model.

Figure 3.8 shows the comparisons of duct temperature between CFD Model and Porous Medium Models. The relative error in Figure 3.8(d) is calculated with the following formula:

$$Relative\ error = \frac{(T_{CFD} - T_{porous\ medium})}{T_{CFD}} \qquad (6)$$

The temperature profiles from different porous medium models show different variation trends. The uniform model obtains hot spots at the corner. That's because the friction at the corner is

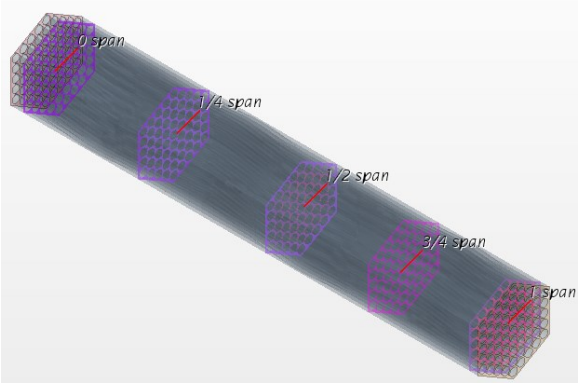
larger, which leads to a lower velocity distribution. In the 3 region mode, cold spots are found at the corner. That's because there is no power density in the outmost region of the 3 region mode. A lower velocity distribution leads to a lower heat transfer. Therefore, less energy is transfer from region 2 to the duct corner. Overall the 3 region model show the best temperature agreement with the CFD calculation because it desscribes the most realistic volumetric heat flux distribution. The relative errors between the 3 region model and the CFD model increase with the elevation. It indicates the error is accumulated along the flow direction.



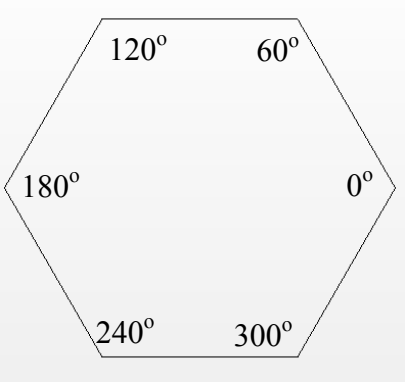
(a) Line_low (b) Line_center

(c) Line_high (d) Relative error

Figure 3.8. Comparisons of duct temperature between CFD Model and Porous Medium Models.

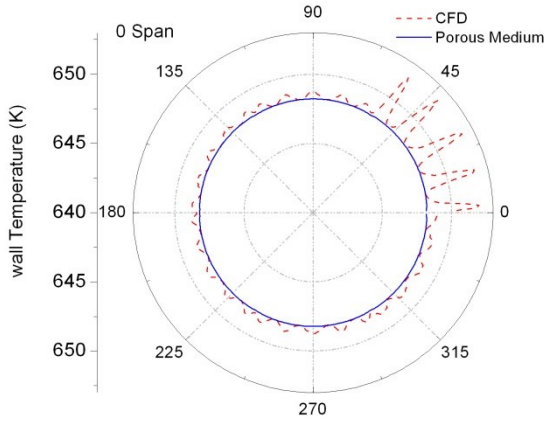


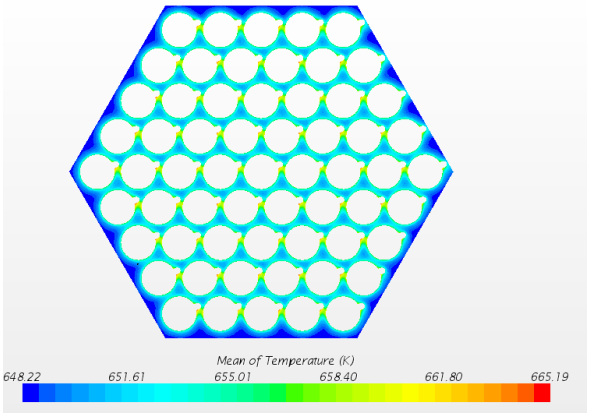
Circumferential Temperature at different elevations



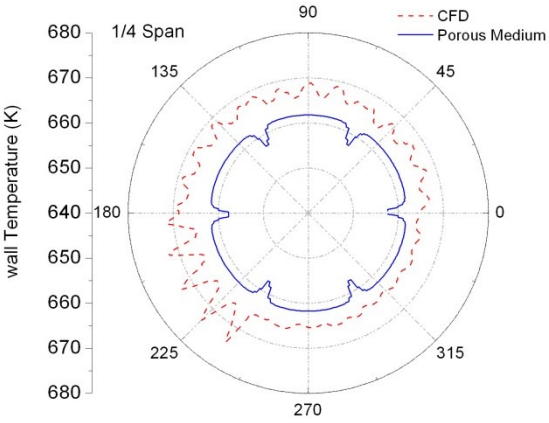
Definition of Angular position

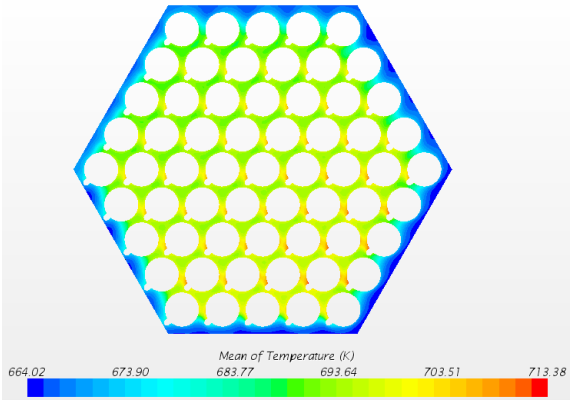
Figure 3.9. Sketch of the span positions and the definition of the angular positions.



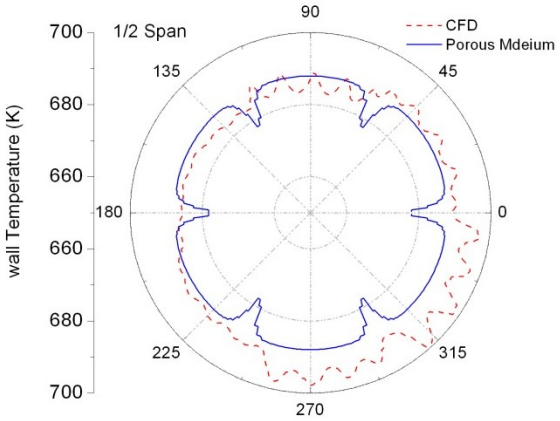


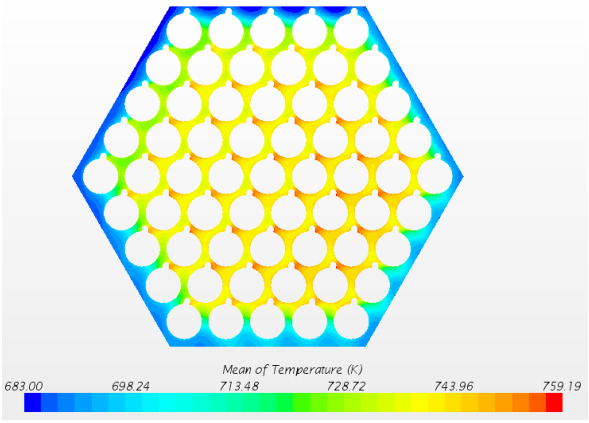
(a) 0 Span



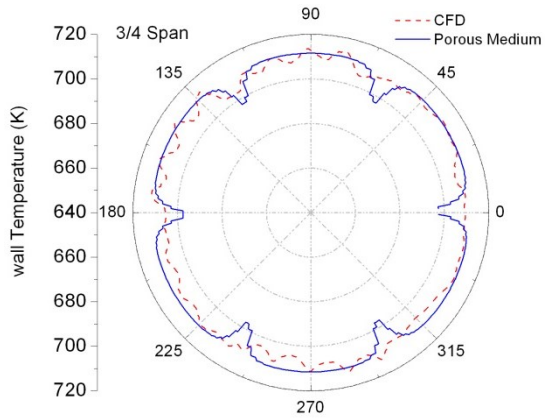


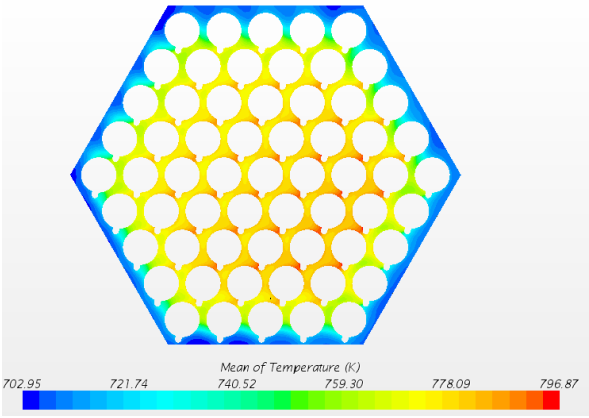
(b) 1/4 Span



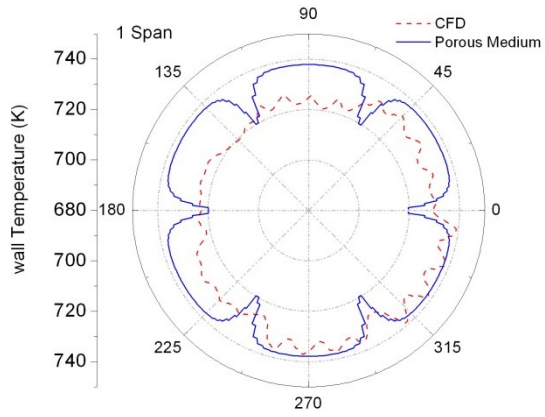


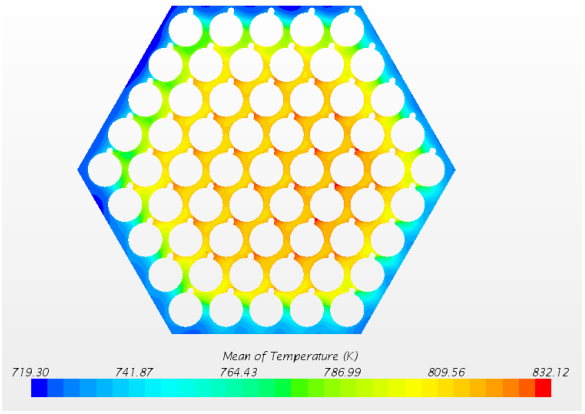
(c) 1/2 Span





(d) 3/4 Span





(e) 1 Span

Figure 3.10 Circumferential duct temperature distributions and the temperature contours at different spans.

In order to estimate the 3 region model porous medium model comprehensively, the circumferential duct temperature at different spans (0 span, 1/4 span, 1/2 span, 3/4 span, 1 span) of the duct are presented in polar coordinate (Figure 3.10). The spans positions and definition of the angular positions is shown in Figure 3.9. The wire wrap pitch is about 0.44 span of the assembly. In Figure 3.10(a), the wire wrap is the main reason for the temperature asymmetry in the CFD results. ;

In fact the porous medium model doesn't take into account the wire wrap effect. In Figure 3.10(b), the wire wrap effect in the CFD model still exists. The porous medium model underestimates the duct temperature because in actuality a part of wire wrap locates at outmost region of the 3 region porous medium model. In Figure 3.10(c), the flow has passed one wire wrap pitch. The wire wrap effect disappears in the CFD calculation. In Figure 3.10(d), the coolant has almost passed 2 wire wrap pitch. Both results are symmetric. The circumferential duct temperature predicted by the porous medium model achieves good agreement with that predicted by the CFD model. In Figure 3.10(e), CFD results become asymmetric again. The discrepancy between the porous medium model and the CFD model becomes greater.

In conclusion, the 3 region model best accounts for the variation in the ratio of solid volume to coolant volume in different parts of the assembly. The results of 3 region porous medium model were found to reproduce the duct temperature of CFD simulation with a good precision. However, the uniform model, which is frequently used in other analyses, exhibited quite poor agreement. Following the results of this study, the 3-region model will become the standard for future work in SHARP with porous media. In further researches, the 3 region porous medium will be applied to 37 pins and 217 pins rod bundle assemblies. *The volumetric heat distribution obtained by PROTEUS will be adjusted according to the local porosity as described in this section to improve the calculation accuracy.*

Future work on this topic will focus on taking into account the wire wraps effect in the porous medium model.

3.2.3 Solid Mechanics Solver (Diablo)

The Diablo code being developed at Lawrence Livermore National Laboratory uses implicit, Lagrangian finite-element methods for the simulation of solid mechanics and multi-physics events over moderate to long time frames [5]. A primary focus is nonlinear structural mechanics and heat transfer. The code provides a venue for applying parallel computation to discretization technologies developed and user-tested in the legacy serial-processor codes NIKE3D and TOPAZ3D. Diablo is built around Fortran 95 data structure objects and a message-passing programming model. The architecture provides flexibility for the addition of other field problems, such as electromagnetics.

In structural analysis of mechanical assemblies, a key functionality is "contact": capturing the interaction between unbonded material interfaces. The Diablo team has broad experience with

contact problems and has created state-of-the-art algorithms for their solution. Their experience with contact motivates the use of low-order spatial discretization, such as eight-node hexahedra for continua and four-node quadrilaterals for shells. Appropriate formulations are employed to accommodate nearly incompressible material models, such as for metal plasticity and rubber elasticity. Global algorithms include second-order and quasi-steady time integration and a number of approaches for nonlinear iteration: full Newton, modified-Newton, multiple quasi-Newton updates, and line search. Linear solvers are utilized from multiple libraries.

3.3 Multi-physics Coupling Methodology

In the future, Diablo, PROTEUS, and Nek5000 will all run simultaneously underneath the CouPE framework and communicate quantities through MBCoupler in MOAB. As an interim step, the coupling has been accomplished through file-based transfer. This is a 2-step process, with the second step consisting of 8 substeps.

1. Individual Nek5000, PROTEUS, and Diablo meshes are generated in the undeformed configuration. Nek5000 and PROTEUS use MOAB mesh files natively. Currently Diablo uses an EXODUS input file and writes the equivalent MOAB (".h5m") file as part of the initialization process. Thus, four mesh files are prepared:
    a. NEK.in.h5m
    b. PROTEUS.in.h5m
    c. DIABLO.in.exo
    d. DIABLO.in.h5m (created by Diablio when it initializes)
2. SHARP iterates the problem until convergence:
    a. Coupled 2-mechanics runs (PROTEUS and Nek5000) are made using the updated mesh
    b. Temperature data from Nek5000 is written to its native "FLD" file format, NEK.temps.FLD
    c. The VisIt utility converts the "FLD" file format to a MOAB ("h5m") file, NEK.temps.h5m
    d. A standalone version of MBcoupler maps the Nek5000 data in .h5m format to the Diablo .h5m file, DIABLO.temps.h5m
    e. Diablo uses the temperature data and coupled solid mechanics to produce deformations, which are written as scalar quantities UX, UY, and UZ to an undeformed MOAB database, DIABLO.disp.h5m
    f. The standalone version of MBCoupler maps the UX,UY,UZ data to the Nek5000 and PROTEUS meshes,
        i. NEK.disp.h5m
        ii. PROTEUS.disp.h5m
    g. The standalone utility DEFORM moves the vertex coordinates on the PROTEUS and UNIQ meshes according to the mapped values of UX, UY, and UZ,
        i. NEK.deformed.h5m
        ii. PROTEUS.deformed.h5m
    h. PROTEUS densities and isotope volume fractions are updated based on the mesh deformation if so desired.

i. The deformed meshes are used as inputs to repeat step 2a above and continue the iterations, as depicted in Figure 3.11.

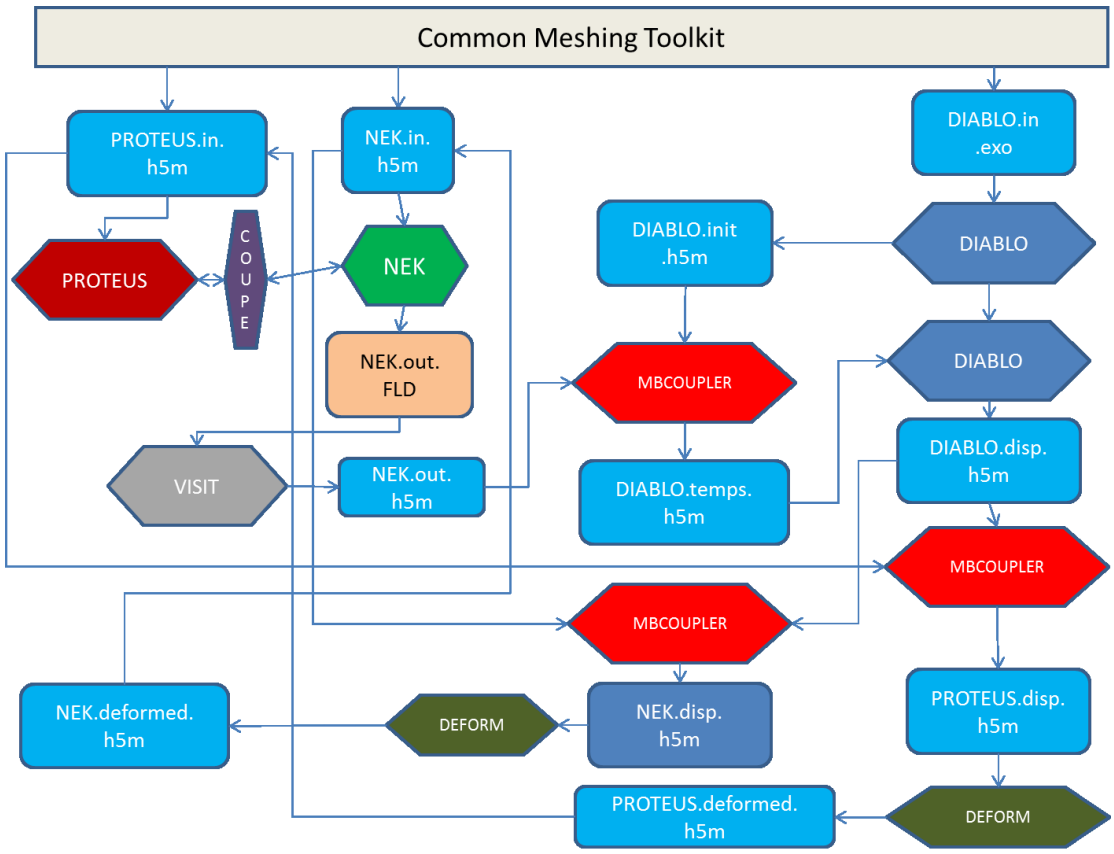


Figure 3.11. Coupling and iteration process.

3.4 Mesh Generation

This demonstration used the reactor geometry (and mesh) generation (RGG) tools in MeshKit for creating the meshed core models. The RGG tools use three-stage methodology comprising three modules –AssyGen, Meshing, and CoreGen. Details of methodology, literature review, handling memory, automation, parallelism, and various results are presented in other papers [25–27] and reports [28–30]. The two key algorithms are contained in AssyGen and CoreGen:

1. AssyGen generates assembly geometries and journal files for meshing of hexagonal and rectangular assembly lattices based on a text-based input file.

2. CoreGen reads an input file describing the reactor core arrangement and generates the reactor core mesh or geometry from its component mesh or geometry files, respectively. It inserts the assemblies into the overall core model, and then merges the matching nodes at the interfaces of assemblies and interstices mesh to form the whole model.

Details on key modeling assumptions, geometry creation, meshing process and salient features of this methodology are presented in the following sub-sections.

Apart from the modeling approximations described in section 3, several meshing restrictions and assumptions were made for modeling the ABTR assemblies, restraint-ring, and full core mesh. PROTEUS/Diablo and Nek5000 impose certain features and required characteristics on the mesh. These were described in [6]. Figure 3.12, Figure 3.13, and Figure 3.14 show some details of the full core mesh.

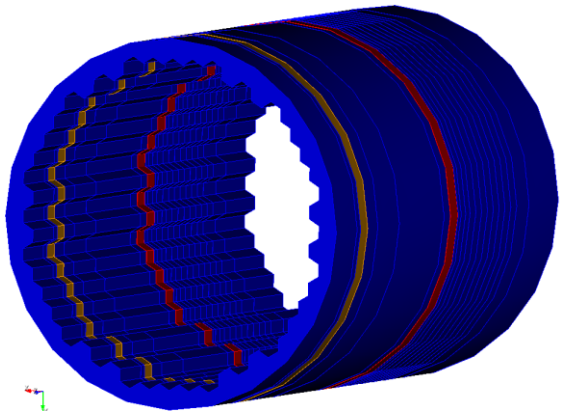


Figure 3.12. Geometry of restraint ring after subtraction of core geometry from the restraint ring cylinders.

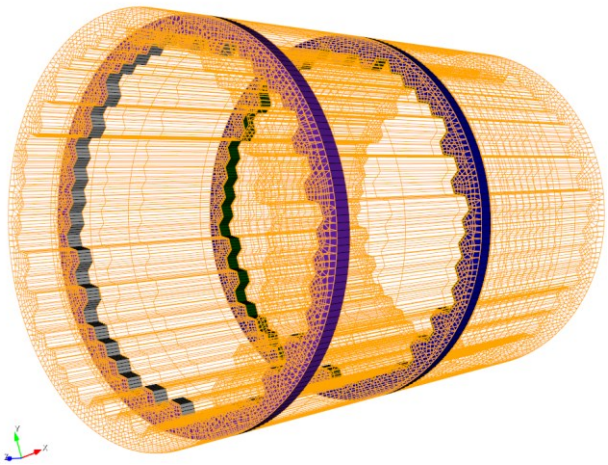


Figure 3.13. Mesh of restraint ring with gap at ACLP and TLP regions (Restraint ring is shown in purple, the gaps at ACLP and TLP are shown in gray and green respectively.)

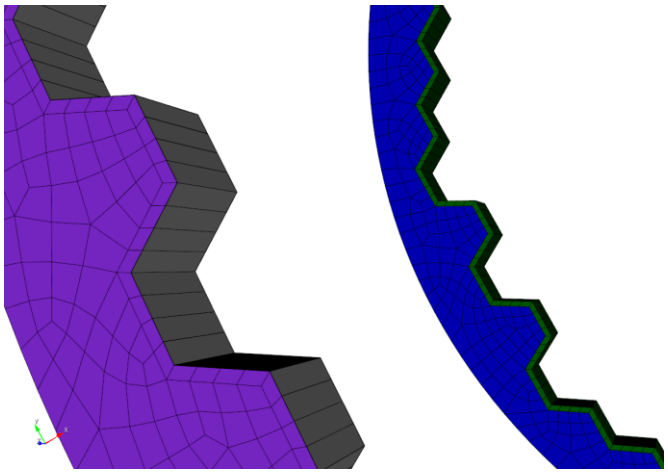


Figure 3.14. Close-up of restraint rings showing the gap at the ACLP (0.0235 cm, gray) and TLP (1.2025 cm, green).

4 Coupled Physics Calculations

In this section preliminary results for the recently carried out full core ABTR calculations are discussed. The standalone physics models have been tested previously as part of a recent milestone. The geometry, mesh, and other inputs have been debugged and confirmed for each of the three physics codes: PROTEUS, Nek5000, and Diablo.

The coupling model used for the demonstration discussed in the following is described in section 3. The strategy can be summarized as follows:

1. at each global iteration of the three physics (SHARP Global iteration) the mesh is updated and the input files are modified with corrected densities
2. as part of each global iteration an inner iteration is performed between PROTEUS and Nek5000.

In the following a simulation of the steady condition of the ABTR at rated power is described.

The full core ABTR problem has 199 assemblies in total, including 60 fuel assemblies. The total power is set to 250 MWt based on the specification. The fuel assemblies comprise three different types: inner core, outer core, and fuel test assemblies, which differ only by fuel composition.

The full core mesh has 825,125 vertices and 789,696 elements. Combined with 48 modeled angles in PROTEUS and 9 energy groups, this problem consists of 356.5 million space-angle-energy degrees of freedom. Parallelization is necessary in order to reduce memory per processor requirements as well as the computer wall-clock time. PROTEUS is highly parallelizable, and 64 processors were used to run the full-core problem. The total wall-clock time using 64 processors was about 27 minutes for the standalone calculation.

The results of an angular convergence study are presented in Table 4.1. The calculation showed little sensitivity in the eigenvalue once a cubature of order L5T7 or higher was used. The L5T7 eigenvalue was 1.00269 in the standalone calculation. Note that the control rods are withdrawn above the active core in this case. In the coupled calculations the eigenvalue is lower both as an effect of the density effects and the core expansion.

Figure 4.1 depicts the power distribution in the fuel assemblies of the full-core ABTR model for the standalone calculation. Figure 4.1a in particular shows the 3D distribution in the entire active core region sectioned along the core centerline to expose interior features. The power distribution is peaked at the radial and axial center of the core, and the z-dependence appears to be approximately cosine shaped. Figure 4.1 b highlights the radial power pattern at Z=138.0 cm (axial midplane of the fuel).

The PROTEUS calculation does not account for gamma heating and therefore has negligible power deposition predictions in the outer regions (reflector and shield). To account for these

effects, power in these regions has been added based on previous calculations [31]. This may lead to inaccuracies – a gamma heating capability in PROTEUS is necessary to avoid this step.

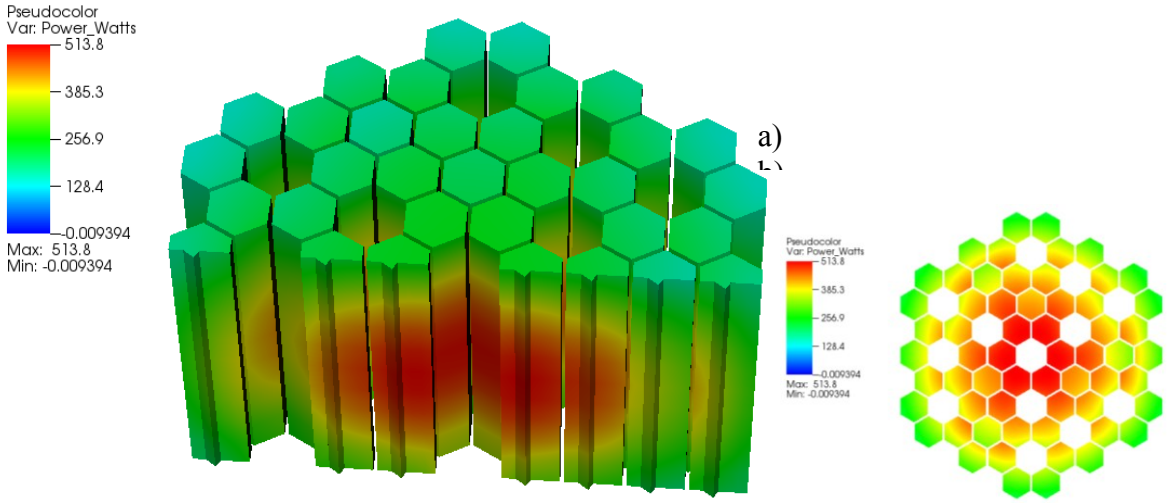


Figure 4.1. Power Distribution for the ABTR full core: a) 3D view, b) cross section at mid active core plane.

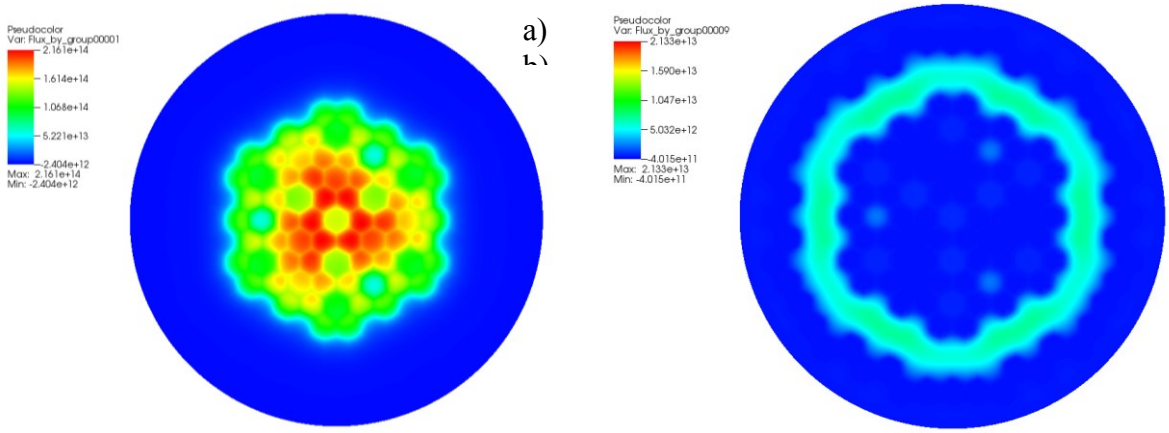


Figure 4.2. Cross section at mid active core plane: a) flux distribution for group 1, b) flux distribution for group 9.

Figure 4.2 shows on the same cross section two of the flux groups.

Table 4.1. Angular convergence – standalone PROTEUS calculation.

Cubature Order	No. of Angles (=2*(L+1)*(T+1))	No. of Angles (PROTEUS) (=(L+1)*(T+1))	K-effective
L1T1	8	4	1.00104
L3T5	48	24	1.00249
L3T7	56	28	1.00239
L3T9	80	40	1.00236
L5T7	96	48	1.00269
L7T7	128	64	1.00277
L7T9	160	80	1.00274

The Nek5000 mesh had 789,696 quadratic elements (polynomial order of N=2). The mesh has several thousands fluid and solid blocks, which were reduced in Nek5000 to 3 types of material: liquid sodium, structure, and solid sodium fill in bypass channels between assemblies. The latter type was used to accelerate convergence by allowing larger time-steps. Otherwise, thin mesh elements in the bypass channel significantly reduce the time-steps due to the Courant-Friedrichs-Lewy constraint when the medium is fluid. This simplification does not affect the final results because flow rate through the bypass assemblies. Inlet boundary and outlet boundary conditions were used I conjunction with the a single region porous medium model (see section 3) in the assemblies. While the single region model is not ideal, modifications of the present simulations to a three region models will be fairly straightforward.

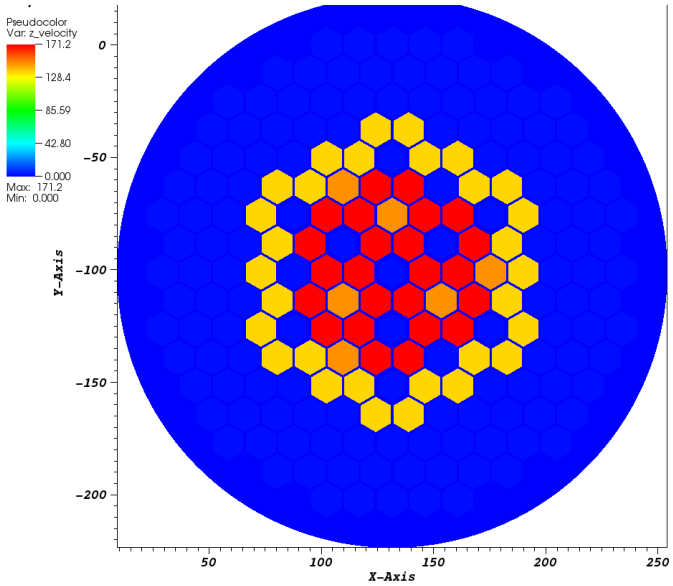


Figure 4.3. Cross section at mid active core plane: Velocity distribution in the z direction [cm/s].

Figure 4.3 shows the velocity distribution at a cross section z=170 cm. The picture is not significantly altered during successive iterations of the coupled solve. Figure 4.4 shows a series of cross sections of the temperature distribution in the core. The steep radial and axial temperature gradients drive the thermal expansion of the core. Figure 4.6 shows a volume rendering of the temperature distribution in the ducts.

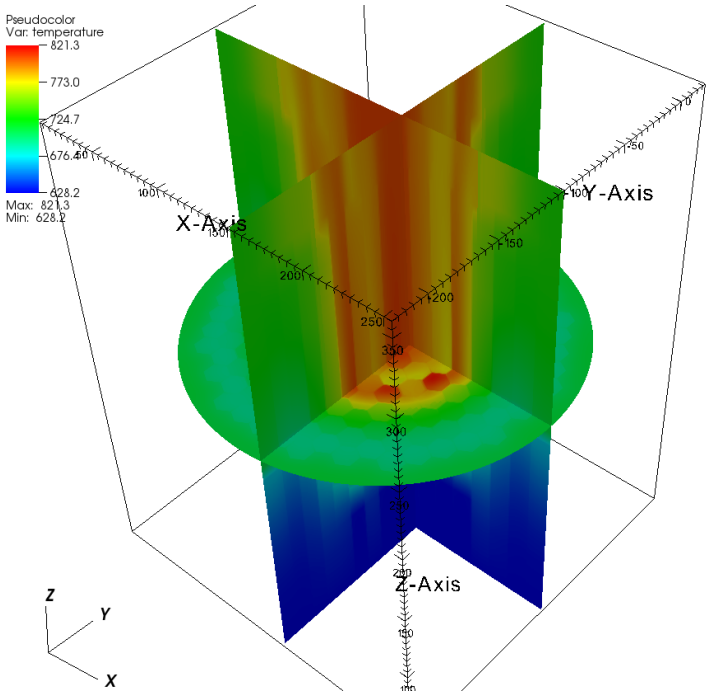


Figure 4.4. Cross section at mid active core plane and vertical cross sections: Temperature [K].

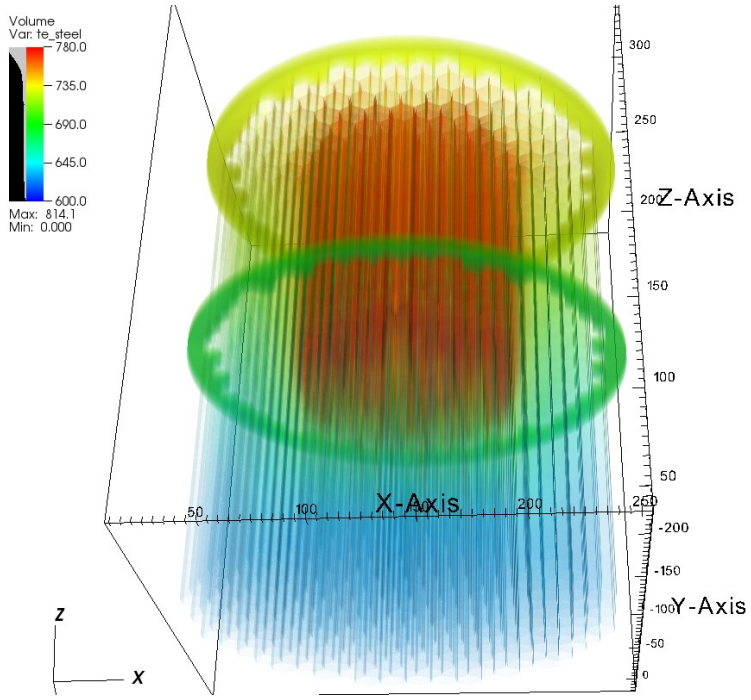


Figure 4.5. Volume rendering of the temperature distribution in the ducts [K].

Figure 4.6 shows the evolution of the k_{eff} as a function of the global iteration. The result should be considered as preliminary as only three iterations have yet been conducted. Experience with a similar geometry shows that at least 5 iterations are needed, but the k_{eff} does not change significantly after the third iteration.

Each result point shown in Figure 4.6 represents the result of a fully converged Nek5000-PROTEUS iteration (50 inner iterations).

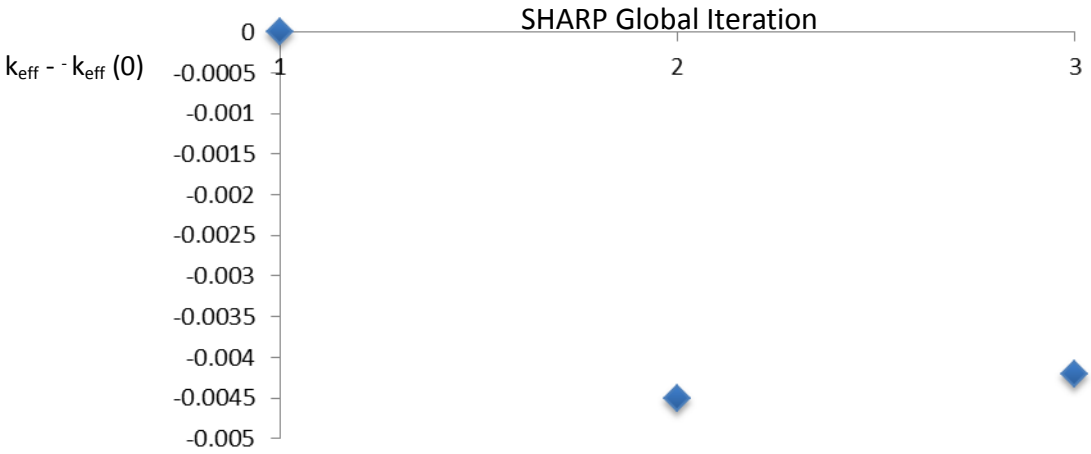


Figure 4.6. k_{eff} as afunction of the global iteration.

Figure 4.7, Figure 4.8, and Figure 4.9 show the displacements computed by Diablo for iteration 2 in direction x and y. We note that the expansion follows a spline shape of the type expected for this power condition.

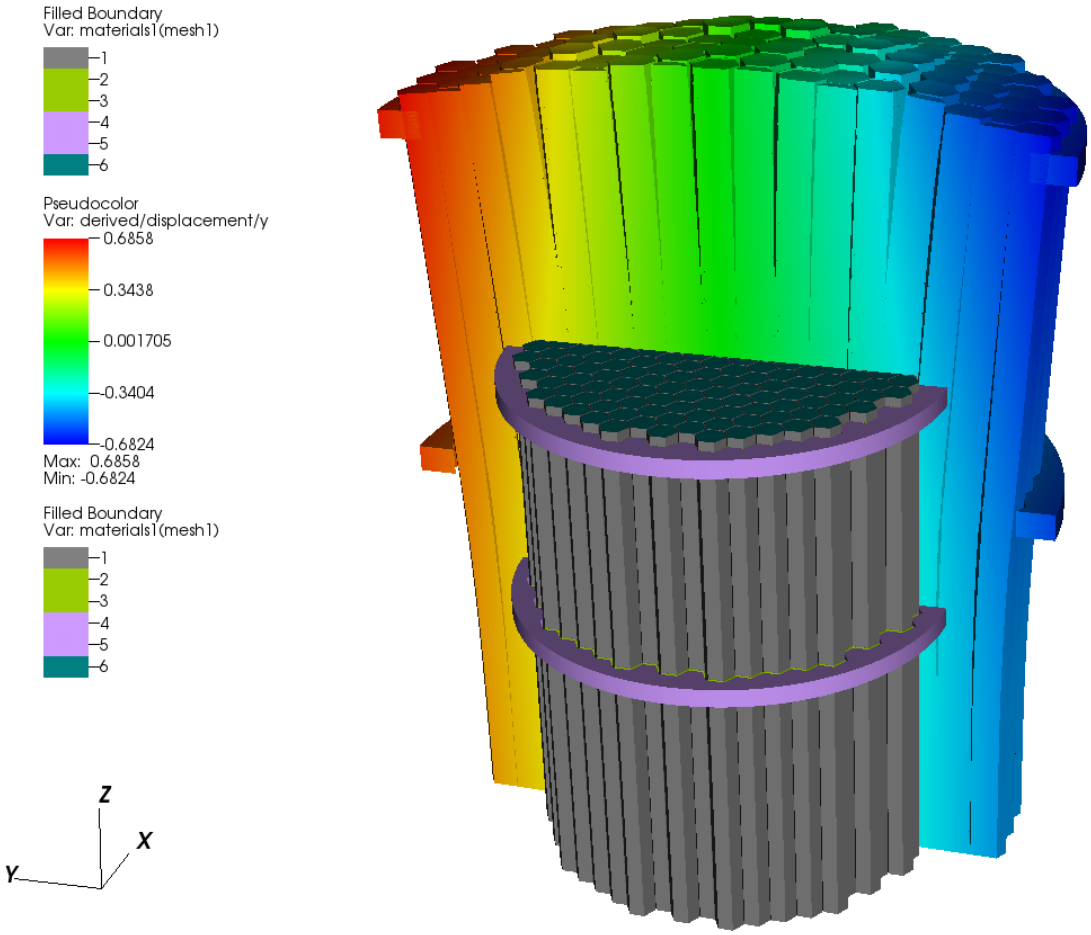


Figure 4.7. Magnified (100x) displacements colored by the displacement in the y direction.

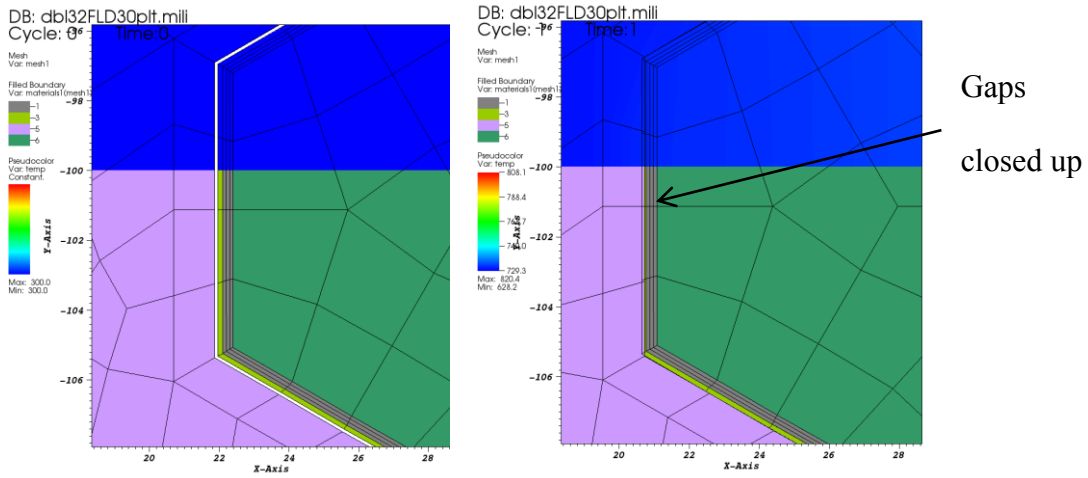


Figure 4.8. Close up on the mesh near the ring, iteration 1 and iteration 2.

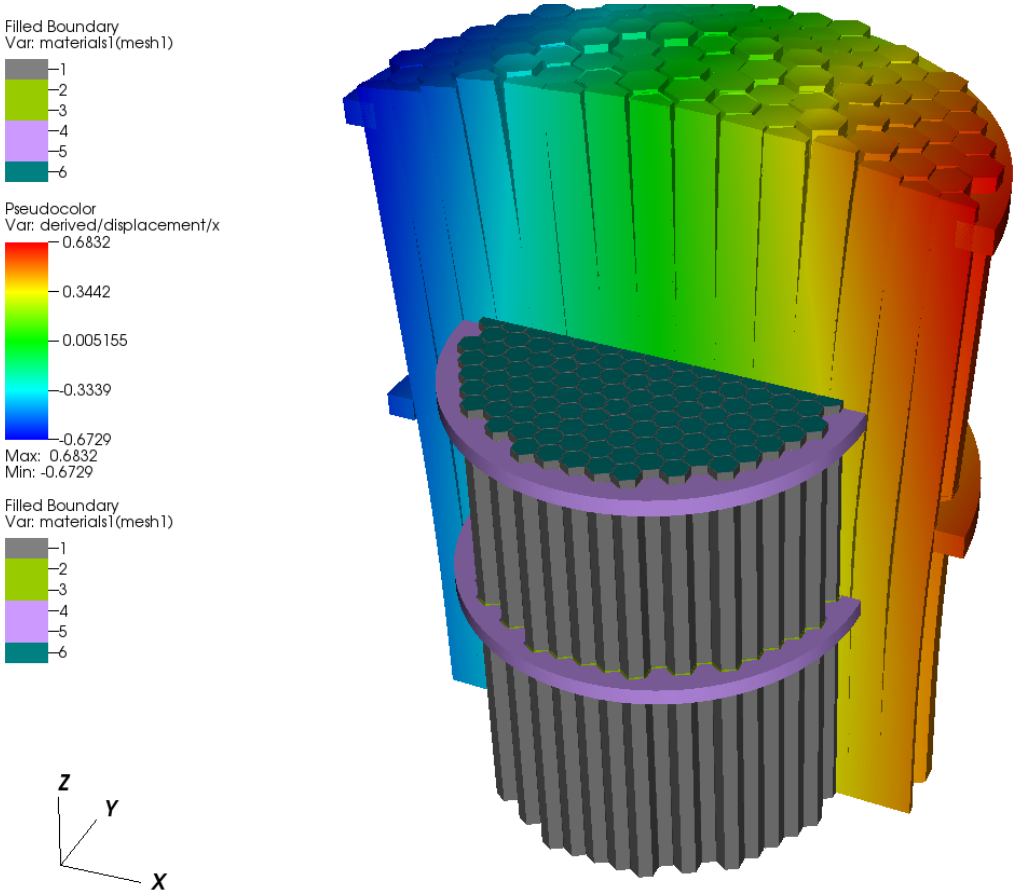


Figure 4.9 Magnified (100x) displacements colored by the displacement in the x direction.

Figure 4.8 shows a close up of the displacement near the TLP ring in iteration 2. It is possible to notice the gap closing and contact between the outermost load pad and the restraint ring.

We note that there is no contact between pads at the ALCP ring, indicating an underestimation of the radial temperature gradient in the outer region (between shield and reflector), further refinement of the model is likely necessary.

5 Conclusions and Future Work

The advanced nuclear reactor modeling and simulation toolkit SHARP was employed to perform a first-of-a-kind analysis of the core radial expansion phenomenon in a full core SFR. Physics models of a full-core model of the ABTR have also been developed for each of the three physics modules. A fully integrated quasi-static simulation of a full core ABTR test problem have also been performed.

For this preliminary demonstration effort, the structural mechanics code Diablo was not yet fully integrated into CouPE, which necessitated off-line mesh mapping and deformation. This process necessitated the development of a utility called DEFORM to modify the thermal-hydraulics and neutronics meshes by applying displacements computed by Diablo. Moreover, isotopics and local material properties (densities) were updated in the PROTEUS input files.

Future work in this area would see additional ABTR core analysis as this capability is extended to support the needs of ARC analysists. Moreover, development efforts in the CouPE framework and Diablo will enable Diablo to be fully integrated, with no need for off-line mesh mapping and deformation. This will greatly facilitate the code coupling process for analysts, and the highly efficient coupling algorithms in CouPE will accelerate simulation development and execution.

6 References

1. A. Siegel, T. Tautges, A. Caceres, D. Kaushik, P. Fischer, G. Palmiotti, M.A. Smith, J. Ragusa, "Software Design of SHARP," in *Proceedings of the Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications (M&C + SNA)*, American Nuclear Society, April 2007.
2. T.J. Tautges, R. Meyers, K. Merkley, C. Stimpson, C. Ernst, *MOAB: A Mesh-Oriented Database*, Sandia National Laboratories report SAND2004-1592, April 2004.
3. M.A. Smith, D. Kaushik, A. Wollaber, W.S. Yang, B. Smith, C. Rabiti, G. Palmiotti, "Recent Research Progress on UNIC at Argonne National Laboratory," in *Proceedings of the International Conference on Mathematics, Computational Methods and Reactor Physics (M&C),* American Nuclear Society, April 2009.
4. P.F. Fischer, J.W. Lottes, S.G. Kerkemier, Nek5000 Web Page, http://nek5000.mcs.anl.gov, 2008.
5. D. Parsons, J.M. Solberg, R.M. Ferencz, M.A. Havstad, N.E. Hodge, and A.P. Wemhoff, *Diablo User Manual*, Lawrence Livermore National Laboratory report UCRL-SM-234927, Sept. 2007.
6. E. Merzari, E. Shemon, J.W. Thomas, A. Obabko, R. Jain, V. Mahadevan, T. Tautges, J. Solberg, R. Ferencz and R. Whitesides , "Multi-Physics Demonstration Problem with the SHARP Reactor Simulation Toolkit", ANL/NE-15/44, Argonne National Laboratory, Dec.2015.
7. E.R. Shemon, J. Grudzinski, C.H. Lee, J. Thomas, Specification of the Advanced Burner Test Reactor Multi-Physics Coupling Demonstration Problem, ANL/NE-15/43, Dec. 2015.
8. T.J. Tautges, H.-J. Kim, A. Caceres, R. Jain, "Coupled Multi-Physics simulation frameworks for reactor simulation: A Bottom-Up approach," in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C)*, American Nuclear Society, Rio de Janeiro, Brazil, May 2011.
9. D. Gaston, C. Newman, G. Hansen, D. Lebrun-Grandi, "MOOSE: a parallel computational framework for coupled systems of nonlinear equations," *Nuclear Engineering and Design*, **239**(10):1768–1778, Oct. 2009.
10. D.E. Keyes et al., "Multiphysics Simulations: Challenges and Opportunities," *International Journal of High Performance Computing Applications*, **27**(1):4-83, 2012.
11. T. Tautges, P. Fischer, I. Grindeanu, R. Jain, V. Mahadevan, A. Obabko, M. Smith, E. Merzari, R. Ferencz, "*SHARP assembly-scale multiphysics demonstration simulations*," ANL/NE-13/9, Argonne National Laboratory, Mar.2013.
12. T.J. Tautges, A. Caceres, "Scalable parallel solution coupling for multiphysics reactor simulation", *Journal of Physics*, Conference Series, **180**, 2009.
13. G.I. Marchuk, *On the theory of the splitting-up method: Volume II of Numerical Solution of Partial Differential Equations*, Academic Press, New York, 1971.
14. D.A. Knoll, D.E. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, **193**(2):357–397, 2004.
15. S. Balay, W.D. Gropp, L. Curfman McInnes, B.F. Smith, "Efficient management of parallelism in object oriented numerical software libraries," in *Modern Software Tools in Scientific Computing*, pp. 163–202, Birkhäuser Press, 1997.

16. M.A. Smith, et al, "UNIC: development of a new reactor physics analysis tool," in *Proceedings of Winter Meeting on International Conference on Making the Renaissance Real*, **97**:565–566, American Nuclear Society, Nov. 2007.
17. Y. Maday, A.T. Patera, "Spectral element methods for the Navier-Stokes equations," in A.K. Noor and J.T. Oden, editors, *State-of-the-Art Surveys in Computational Mechanics*, pp. 71–143, ASME, New York, 1989.
18. A.G. Tomboulides, J.C.Y. Lee, and S.A. Orszag, "Numerical simulation of low Mach number reactive flows," *Journal of Scientific Computing*, **12**:139–167, June 1997.
19. A.G. Tomboulides, M. Israeli, G.E. Karniadakis, "Efficient removal of boundary-divergence errors in time-splitting methods," *Journal of Scientific Computing*, **4**:291–308, 1989.
20. *STAR-CD v4.12 Methodology Guide*, CD-adapco, Ltd., 2011.
21. E.U. Khan, W.M. Rohsenow, A.A. Sonin, N.E. Todereas, A Porous Body Model for Predicting Temperature Distribution in Wire-Wrapped Fuel Rod Assemblies, *Nuclear Engineering and Design*, **35**, pp. 1-12 (1975)
22. W.D. Pointer, P. Fischer, A. Siegel and J. Smith, "RANS-based CFD Simulations of Wire- Wrapped Fast Reactor Fuel Assemblies," *Proceedings of ICAPP'08*, Anaheim, CA (2008).
23. W.D. Pointer, J.W. Thomas, T.H. Fanning, et al., "RANS-Based CFD Simulations of Sodium Fast Reactor Wire-Wrapped Pin Bundles," *Proceedings of M&C 2009*, Saratoga Springs, New York (2009).
24. R. Hu and T.H. Fanning, "Development of a Three-Dimensional Momentum Source Model for Wire-Wrapped Rod Bundles", *Proceedings of NURETH-14*, Toronto, Canada, September 25-30, (2011).
25. R. Jain, T.J. Tautges, "NEAMS MeshKit," presented at International Congress on the Advances in Nuclear Power Plants, Chicago, 2014.
26. T.J. Tautges, R. Jain, "Creating geometry and mesh models for nuclear reactor core geometries using a lattice hierarchy-based approach," *Engineering with Computers*, **28**(4):319-329, 2011.
27. R. Jain, T.J. Tautges, "RGG: Reactor Geometry (and Mesh) Generator," presented at International Congress on the Advances in Nuclear Power Plants, Chicago, 2012.
28. R. Jain, T.J. Tautges, "*MeshKit"*, ANL/MCS-TM/336, Argonne National Laboratory, 2013.
29. R. Jain, "Report on FY11 Extensions to MeshKit and RGG", ANL/MCS-TM-316, Argonne National Laboratory, Sep. 2011.
30. T.J. Tautges, R. Jain, "Mesh Copy/Move/Merge Tool for Reactor Simulation Applications" DOE Reactor Campaign, April 30, 2010.
31. Y.I. Chang, P.J. Finck, C. Grandy, *Advanced Burner Test Reactor Preconceptual Design Report*" Argonne National Laboratory report ANL-ABR-1 (ANL-AFCI-173), Sept. 2006.

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov